

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

Jeff Sonntag and John Stonick
Synopsys, Inc.
2025 NW Cornelius Pass Road
Hillsboro, OR 97124

Abstract - In this 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 VCO in a typical analog 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. Finally, measured results are presented that corroborate the modeled results.

I. INTRODUCTION

Multi-Gigabit per second (Gbps) serial binary links are fast replacing traditional parallel data links in many applications. Examples include PCI moving towards PCIe and SATA moving towards SATA. Additionally, there exist many other applications with multi-Gbps serial links such as 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 ASIC, rather than the core function of a mixed signal ASSP. For this reason, effective solutions must be extremely low in power, implementable in the cheapest of digital process technologies, 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 CDR's 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.

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 PLL, including a bang-bang phase detector,

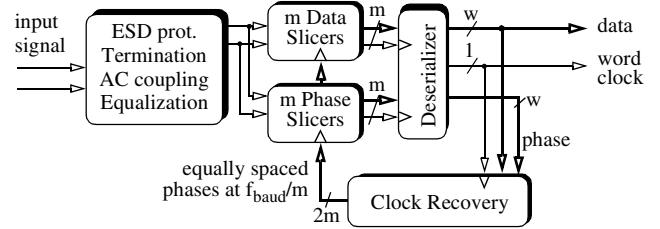


Fig. 1. Typical receiver and CDR

Charge Pump Loop Filter (CPLF) and a Voltage Controlled Oscillator (VCO) as shown in Fig. 2 [1][2][3].

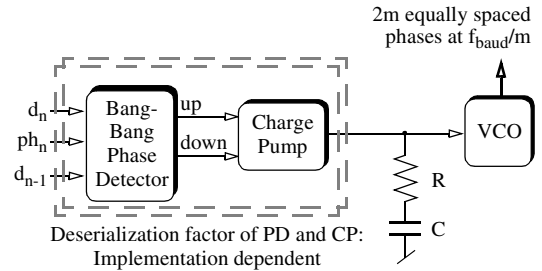


Fig. 2. Analog Clock Recovery Unit

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. The Bang-Bang Phase Detector

The bang-bang phase detector is common to many analog CDRs and the digital CDR proposed here. It produces a non-zero output of either +1 or -1 for data transitions and a zero output for non-transitions. 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 noise.

The 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 grey phase sample in Fig. 3.

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	

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_{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 $1 - 2 \cdot Q(V_{DC}/\sigma_v)$, 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. For small values of (V_{DC}/σ_v) , this may be approximated as a straight line:

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

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

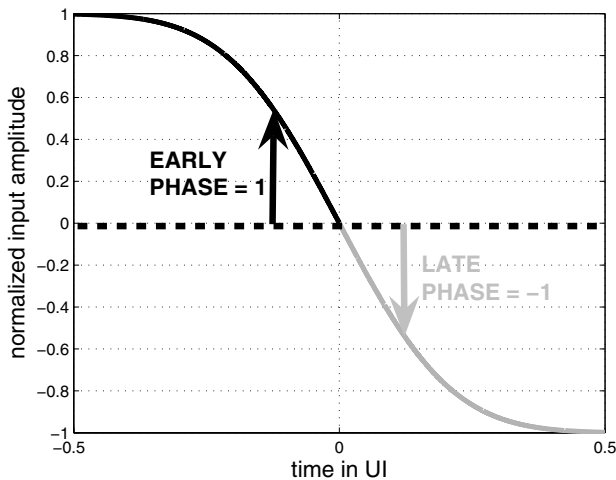


Fig. 3. Early-late phase sampling

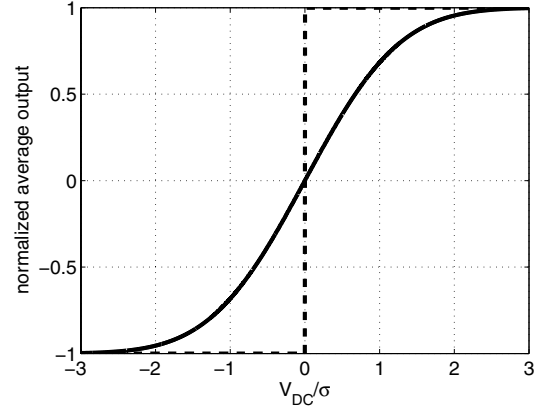


Fig. 4. Average output of ideal slicer, as function of mean input

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 $e \cdot \text{slope}$:

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

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:

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

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

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

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 non-ideal 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

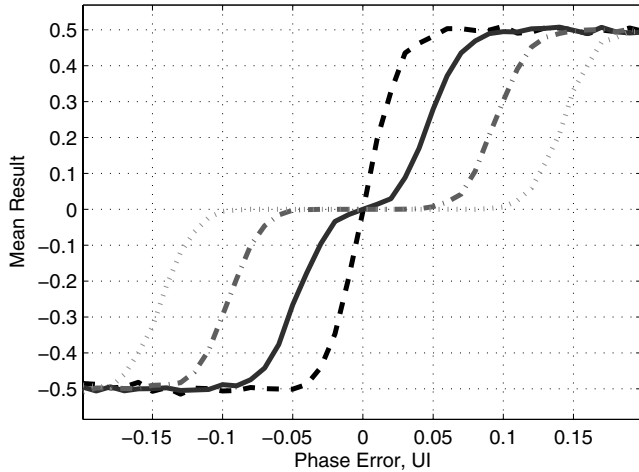


Fig. 5. Simulated Phase Detector Transfer Functions. Phase slicer offsets are: $\{0, 0.15, 0.30, 0.45\} * A$; $\sigma_j = 0.06 * A$.

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 non-ideal 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 equation (3).

D. Linearized Small Signal Model

There have been many excellent papers on the design and analysis of this type of CDR system [1][2][4][5]. A linearized model is shown in fig. 6.

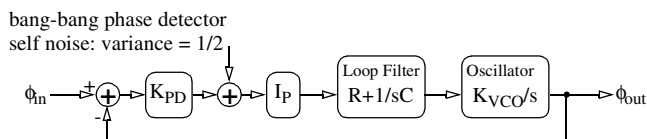


Fig. 6. Linearized Model of Analog Clock Recovery Unit

The loop gain for the linearized system is given by (5).

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

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 $1/K_{PD}$ 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 reflected input 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][8][9][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'll designate it as the *word* rate. Operating at this lower rate has costs (latency), but makes the required computations both possible and power and area inexpensive. Decimation is described in subsection 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 DLLs, phase mixers/interpolators, and PLLs [7][8][9][10][11][12][13]. Implementation of the DPC is not covered in this paper.

A. Analogy to Analog Implementation

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

$$\left(\frac{K_{VCO}}{s}\right)\left(I_p R + \frac{I_p}{sC}\right) \rightarrow \left(\frac{TK_{VCO}}{1-z^{-1}}\right)\left(I_p R + \frac{TI_p}{(1-z^{-1})C}\right) \quad (6)$$

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

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

By comparing (6) and (7) we can see that the **phase update gain** (*phug*) models the proportional path gain in the CPLF, that the **frequency update gain** (*frug*) models the integral path gain in the CPLF and that K_{DPC} models the gain of the VCO, K_{VCO} . The extra term, $z^{-N_{EL}}$, 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, $T_{word} \cdot N_{EL}$, is not controlled and is allowed to approach $1/(4 \cdot f_{unity\ gain})$ a severe loss in phase margin occurs. Design techniques which minimize N_{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 trade-offs 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 trade-offs while providing more detail on the blocks in fig. 7.

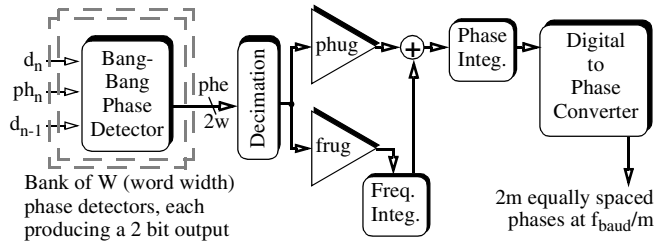


Fig. 7. Digital PLL Architecture

B. Decimation by Voting

The most straightforward approach to the decimation operation is by the use of an 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.

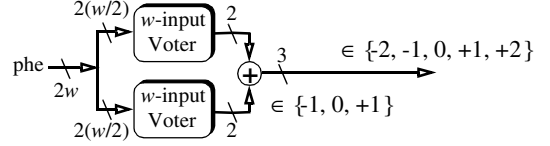


Fig. 8. Faster Decimation with Voting

Decimation via boxcar filter produces a DC gain corresponding to the decimation factor, w . Decimation via voting has a reduced gain 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 reflected 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.

Decimation by voting across four inputs has a gain which is 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.

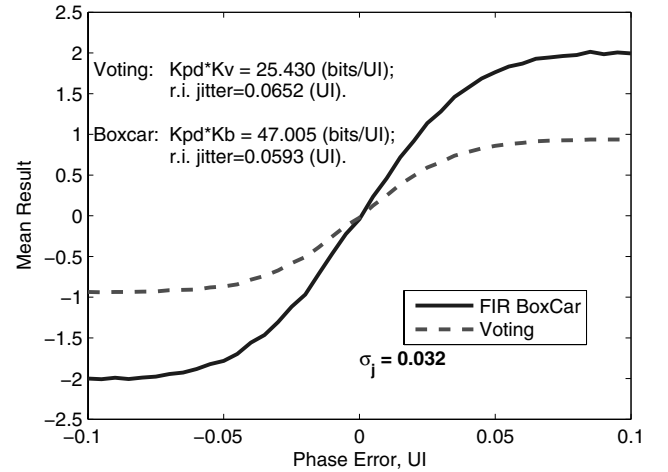


Fig. 9. Simulated Decimation by Voting and Boxcar FIR

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 are given in the following table with some description being given after the table.

In the model, the element K_{PD} is the phase detector gain as given in (4). To get a meaningful value we will use the jitter of 7.5ps observed from the measured results provided later in

TABLE II
TEST DEVICE DIGITAL CDR PARAMETERS

Parameter	Value
K_{DPC}	1 UI / 2^9 bits
K_V	$8 \cdot 54 = 4.32$
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

the paper. To use this jitter value in (4) it must be converted into radians. For 5Gbps operation the period is 200ps. Thus, σ_j is $7.5/200 \cdot 2\pi$ radians. When this is substituted into (4) we get the value in the table.

The next element in the model is K_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 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_{DPC} is the gain through the DPC. This corresponds to the resolution of the DPC in units of UI per bit. The resolution of the DPC is a trade-off between the truncation noise induced by low resolution and the complexity and power required for high resolution.

Finally, recall that the term $z^{-N_{EL}}$ 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, Φ_{in}/Φ_{err} and the transfer function, Φ_{samp}/Φ_{in} . To compute either of these it is beneficial to first compute the loop gain, $L(z^{-1})$, from Φ_{err} to Φ_{samp} .

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

The jitter transfer function is proportional to the reciprocal of the phase error transfer function and is given (9).

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

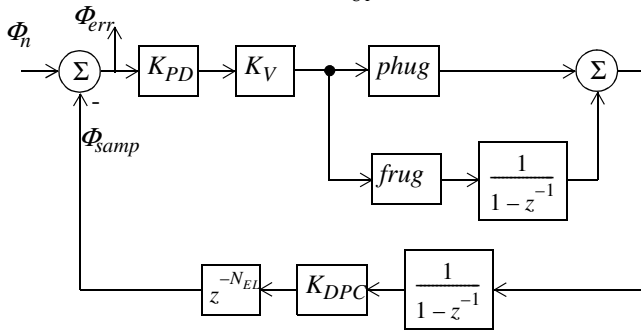


Fig. 10. Linearized Model of Proposed Architecture

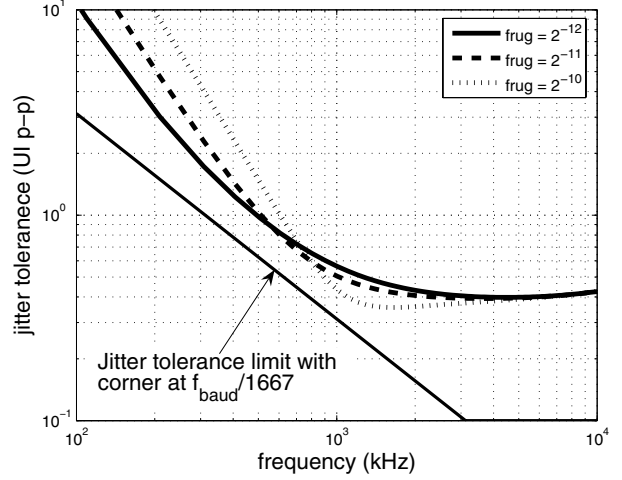


Fig. 11. Calculated jitter transfer function

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 T_{UI} is 200ps and the observed jitter was 7.5ps 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.

$$\Phi_{samp}/\Phi_{in} = (L(e^{-j\omega}) / (1 + L(e^{-j\omega}))) \quad (10)$$

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

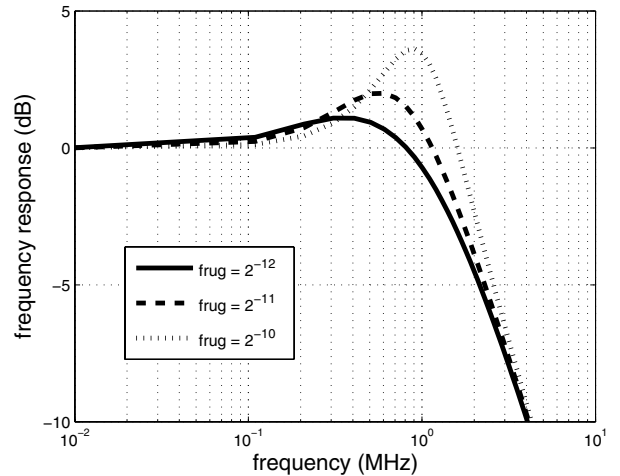


Fig. 12. Calculated phase transfer function

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 non-saturating to allow the phase to move more than 1 UI. The frequency integrator is signed and saturating since it is used to track both +/- ppm offsets. Saturation is required because we do not want the frequency register to “roll over” from large positive values to large negative values. Finally, in the implementation, the phase and frequency integrators are fed from the sum of 2 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

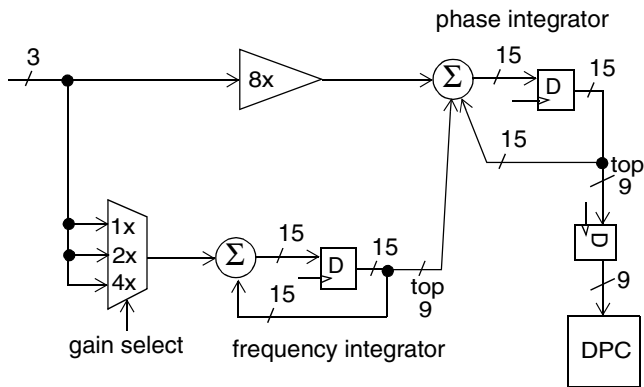


Fig. 13. Sample Realization

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 + 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 $(3.98/(8*512))*10^6 = 972\text{ppm}$. 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 $972/2^8=3.8\text{ppm/lb}$.

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

IV. MEASURED RESULTS

The measured results were obtained using a CDR integrated on a 0.13um CMOS test device (the details of which can be found in [11]) operating at 5 Gbps over a 34” FR-4 backplane trace through two connectors. The data pattern was generated using a 31st order primitive polynomial.

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.

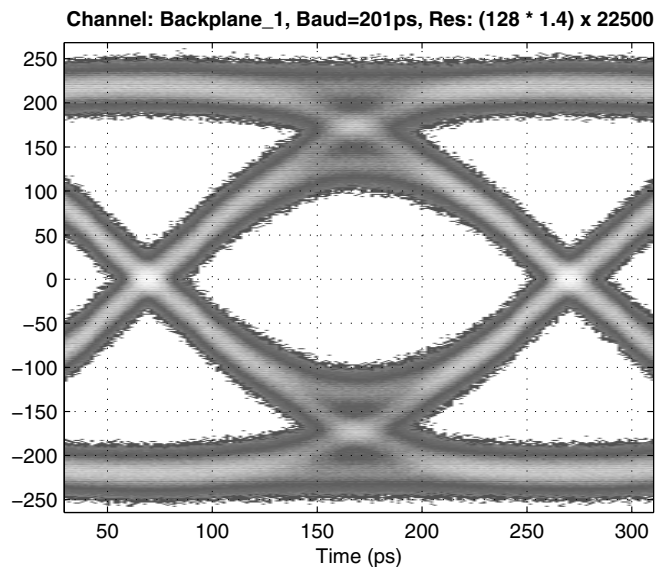


Fig. 14. Five Gbps receive signal used in obtaining measured results

A micrograph of the chip is shown in fig. 15.

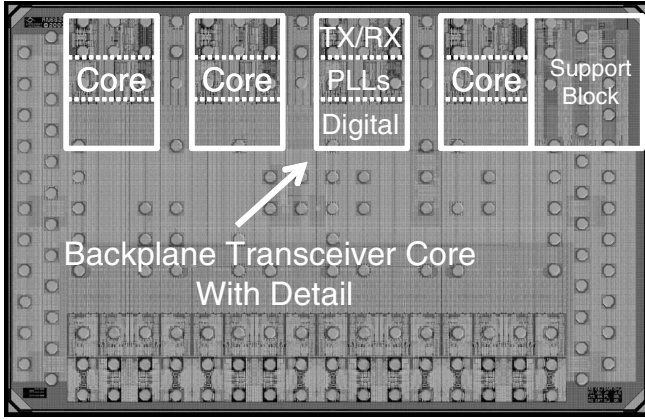


Fig. 15. Micrograph of test chip

A. $K_{PD} * K_V$ 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_{pd} . The results agree well with the simulated results of fig. 5.

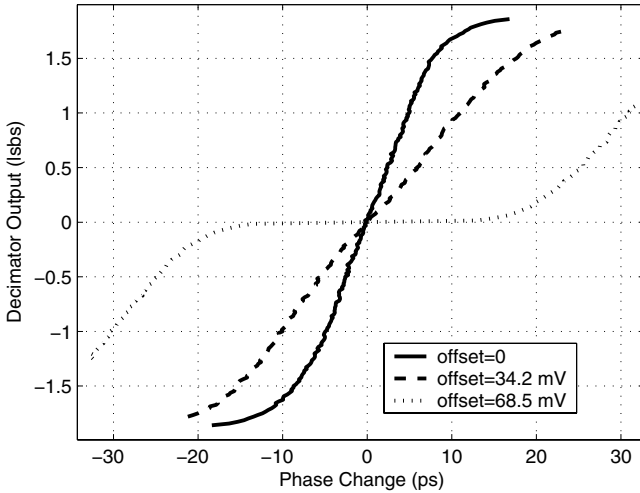


Fig. 16. Measured combined phase detector and decimator transfer function.

B. Jitter Tolerance Measurement

Jitter tolerance was measured by determining the pk-pk amplitude of jitter at each frequency which could be tolerated in order to produce a BER of approximately $1e-10$. This was repeated for the three values of *frug*. The results are plotted in

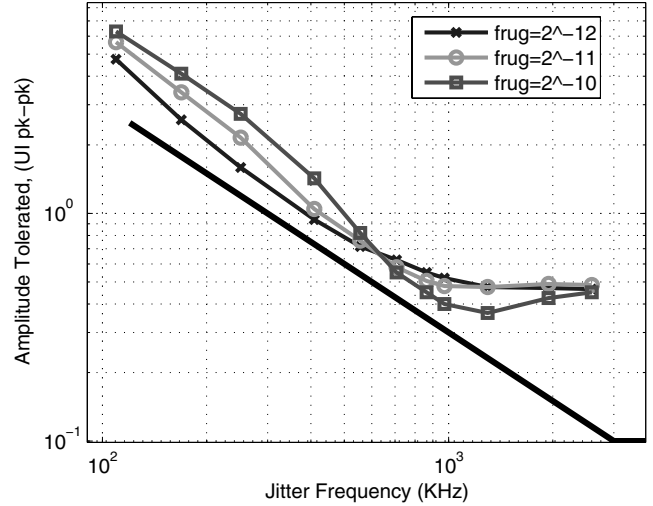


Fig. 17. Measured jitter tolerance

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.

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-Gb binary data links.

Several 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 doesn't need to make the same power/noise trade-off required of the analog PLL; 2.) insensitivity to long runs of transition-free data patterns; 3.) invariance of characteristics over PVT; 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; 7.) production testing of logic gates is much more straightforward than analog circuits; 8.) ease of adding bench test hooks for characterization; 9.) ease of allowing flexible control of design parameters.

In conclusion, DPLL-based CDR's are area and power efficient and provide flexible, effective functionality for Gpbs data links.

ACKNOWLEDGMENTS

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 for 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 PLL's for clock and data recovery in serial data transmission systems," in *Phase-Locking in High-Performance Systems*, B. Razavi, Ed: IEEE Press, 2003, pp.34–45.
- [2] M. Ramezani, C. Andre, and T. Salama, "A 10Gb/s CDR with a half-rate bang-bang phase detector," *Proceedings of the 2003 International Symposium on Circuits and Systems*, vol.2, May 2003, pp. 181-4.
- [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, March 2005, pp. 735-743.
- [4] Jri Lee; Kundert, K.S., Razavi, B, "Analysis and modeling of bang-bang clock and data recovery circuits," *IEEE J. of Solid-State Circuits*, vol. 39, no. 9, Sept. 2004, pp. 1571-80.
- [5] M. Ramezani, C. Andre, and T. Salama, "Jitter analysis of a PLL-based CDR with a bang-bang phase detector," *The 2002 45th Midwest Symposium on Circuits and Systems*, Volume 3, vol.3, Aug. 2002, pp.393-6.
- [6] D. P. Atherton, *Nonlinear Control Engineering*, Van Nostrand Reinhold Co., London & New York; full edition 1975, student edition 1982.
- [7] H. Takauchi, *et al*, "A CMOS multichannel 10-Gb/s transceiver," *IEEE J. of Solid-State Circuits*, vol. 38, no. 12, Dec. 2003, pp. 2094 - 2100.
- [8] H. Tamura, *et al*, "5-Gb/s bidirectional balanced-line link compliant with plesiochronous clocking," *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, no. 375, Feb. 2001, 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. of Solid-State Circuits*, vol. 39, no. 9, Sept. 2004, pp. 1553 - 1561
- [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, May 2003, pp. 747-753.
- [11] K. Krishna. *et al*, "A 0.6 to 9.6Gb/s Binary Backplane Transceiver Core in 0.13 μ CMOS," *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 65-65, Feb. 2005.
- [12] J. Parker, *et al*, "A 15mW, 3.125GHz PLL for Serial Backplane Transceivers in 0.13mm CMOS," *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, pp. 412-413, Feb. 2005.
- [13] S. Sidiropoulos and M. A. Horowitz, "A semidigital dual delay-locked loop," *IEEE J. of Solid-State Circuits*, Nov. 1997, pp. 1683-1692.
- [14] J. Yueming, A. Piovaccari, "A compact phase interpolator for 3.125G Serdes application," *2003 Southwest Symposium on Mixed-Signal Design*, Feb. 2003, pp. 249 - 252.