An Embedded All-Digital Circuit to Measure PLL Response

Dennis M. Fischette, *Member, IEEE*, Alvin L. S. Loke, *Senior Member, IEEE*, Richard J. DeSantis, and Gerry R. Talbot, *Member, IEEE*

***Abstract*—We present an all-digital measurement circuit that enables wafer-level test and characterization of phase-locked loop (PLL) response. Through modifications only in the PLL feedback divider state machine, this technique facilitates accurate estimation of PLL frequency-domain closed-loop bandwidth and gain peaking by respectively measuring the time-domain crossover time and maximum overshoot of phase error to a self-induced phase step in the feedback clock. These transient measurements are related back to bandwidth and peaking through the proportionality relationships of crossover time to reciprocal bandwidth and maximum overshoot to peaking. The design-for-test circuit can be used to generate a transient plot of step response, measure static phase error, and observe phase-lock status. We report silicon results from two demonstration vehicles built in a 45-nm SOI-CMOS logic technology for high-performance microprocessors.**

***Index Terms*—Bandwidth, CMOS integrated circuits, design-for-test, embedded test, loop response, measurement circuitry, peaking, phase-locked loops.**

I. Introduction

MANY high-speed wireline applications such as PCI Express® require a phase-locked loop (PLL) to produce a low-jitter clock at a given frequency while meeting stringent jitter modulation bandwidth and gain peaking requirements. For example, the PCI Express 2.0 standard, which specifies a link rate of 5 Gb/s per lane, calls for either 5–8 MHz bandwidth with peaking below 1 dB or 8–16 MHz bandwidth with peaking below 3 dB [1]. Locktime requirements, which dictate exit times from power-saving standby states, may further restrict PLL bandwidth from being too low.

Wafer manufacturing process variations in the transistors and passive elements as well as operating supply voltage and temperature (PVT) variations make guaranteeing a narrow range of PLL response difficult. For example, loop parameters such as voltage-controlled oscillator (VCO) gain (K_{VCO}) may vary by more than 3× across PVT design corners, corresponding to a 3× spread in bandwidth. The primary motivation of this work is to ensure that parts comply with specification using embedded design-for-test capability. The secondary motivation is to re-program parts that do not meet specification to recover product yield wherever possible.

PLL loop response is conventionally specified by a closed-loop transfer function of phase modulation. Fig. 1 illustrates the transfer functions of two PLLs with identical reference clock and output frequencies. The transfer function shows, as a function of the modulation frequency (f_m), how the PLL responds to a phase-modulated reference clock. Here, modulation frequency is not to be confused with the reference input frequency itself (f_{ref}). The phase modulation can be intentional, such as the case of spread-spectrum modulation to reduce electromagnetic interference [2], or unintentional, in which case it is regarded as input noise contributing to PLL output jitter [3]. For an input phase

$$\phi_{\mathrm{in}}(t) = 2\pi f_{\mathrm{ref}} t + A_{\mathrm{in}} \sin(2\pi f_m t) \tag{1}$$

where A_{in} is the input phase modulation amplitude, the PLL output phase (for relatively small modulation) will be given by

$$\phi_{\mathrm{out}}(t, f_m) = N \cdot (2\pi f_{\mathrm{ref}} t - \phi_{\mathrm{offset}}) + A_{\mathrm{out}}(f_m) \sin(2\pi f_m t + \phi_m(f_m)) \tag{2}$$

where N is the feedback divisor, ϕ_{offset} is the PLL input static phase offset, and $A_{\mathrm{out}}(f_m)$ and $\phi_m(f_m)$ are respectively the magnitude and phase response functions of the reference clock modulation. The resulting magnitude response of the PLL transfer function is therefore

$$H(f_m) = \frac{1}{N} \frac{A_{\mathrm{out}}(f_m)}{A_{\mathrm{in}}} \tag{3}$$

which is normalized by the feedback divisor (N) to account for the frequency- (and phase-) multiplying action of the PLL. A PLL behaves as a low-pass filter of reference modulation since its output follows the reference at low modulation frequencies but cannot track higher modulation frequencies.

PLL loop response is summarized by its bandwidth and peaking characteristics. The PLL bandwidth, measured at the −3 dB point in the Fig. 1 curves, is chosen by balancing the effects of reference input noise and internally generated PLL noise to achieve the lowest PLL output clock jitter. Lower bandwidths attenuate more noise in the reference input spectrum at the expense of rejecting less noise generated by the PLL circuitry while higher bandwidths achieve the opposite. Bandwidth targeting is a system consideration that depends on the quality of the selected off-chip reference source and downstream on-chip reference clock distribution as well as the performance of the PLL given its design constraints, notably

Manuscript received November 24, 2009; revised February 16, 2010; accepted March 25, 2010. Current version published July 23, 2010. This paper was approved by Guest Editor Anthony Chan Carusone. This work was supported by Advanced Micro Devices (AMD).

D. M. Fischette and R. J. DeSantis are with Advanced Micro Devices, Inc., Sunnyvale, CA 94085-3905 USA (e-mail: dennis.fischette@ieee.org).

A. L. S. Loke is with Advanced Micro Devices, Inc., Fort Collins, CO 80528-3419 USA (e-mail: alvin.loke@ieee.org).

G. R. Talbot is with Advanced Micro Devices, Inc., Boxborough, MA 01719-1200 USA.

Digital Object Identifier 10.1109/JSSC.2010.2048143

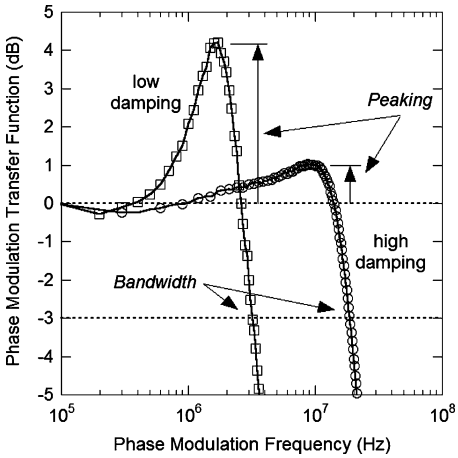


Fig. 1. Examples of measured PLL phase modulation transfer functions.

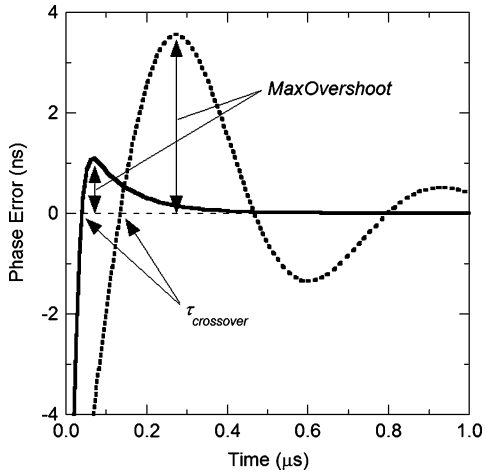


Fig. 2. Simulated PLL transient response to input phase step.

power consumption and area. The maximum value in the transfer function is referred to as the gain peaking. Higher peaking is undesirable from a jitter perspective since the PLL amplifies phase modulation around the peaking frequency but can be beneficial to reduce step response time. In Fig. 1, one PLL exhibits large peaking and low bandwidth while the other shows little peaking but high bandwidth. Similar differences often result from PVT variations although this example is more extreme than usual.

The PLL closed-loop transfer function is often measured on a test bench using a sinusoidal signal generator to modulate the reference clock and an oscilloscope or spectrum analyzer to measure the PLL response. For example, the transfer functions in Fig. 1 were obtained by modulating a 100 MHz reference clock at various frequencies (one modulation frequency at a time) and observing the amplitudes of the resulting output reference spurs on a spectrum analyzer. (Reference spurs are symmetric sideband tones offset by the modulation frequency from the PLL output carrier.) This technique may require many seconds, sometimes even minutes, to complete. In the production test environment, tester time is very expensive. Also, since traditional methods often require driving the high-speed PLL output clock off-chip, an unachievable requirement for wafer-level testing, a part may need to be packaged before its PLL response can be measured. This escalates the cost implication of packaging parts that are not known good dies, especially with the increasing integration of processor cores in costly multi-chip module packages [4]. These realities motivate the need for a faster, less expensive, and on-chip technique to measure PLL loop response [5]–[9].

II. Measurement Theory

The proposed measurement circuit performs time-domain measurements of PLL output phase in response to an induced phase step. These measurements are fundamentally correlated to bandwidth and peaking in the frequency domain. The phase of the reference clock is instantaneously advanced (alternatively, the phase of the feedback clock is instantaneously retarded) and the resulting PLL phase error transient is recorded as shown in Fig. 2. Similar to other second-order feedback systems, the PLL tends to overcorrect or overshoot as it eliminates

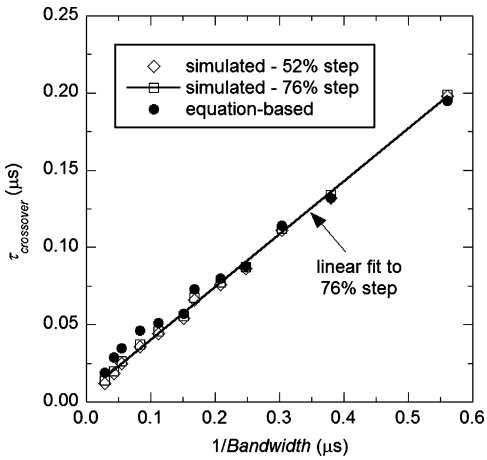


Fig. 3. Simulated and equation-based $\tau_{\mathrm{crossover}}$ versus reciprocal bandwidth for two different input phase steps.

the induced phase error. If the PLL is underdamped, the PLL will ring several times before settling to its final locked state. Even an overdamped PLL will exhibit some overshoot due to the presence of unavoidable parasitic poles in any real-world PLL transfer function. A key metric in the PLL step-response is $\tau_{\mathrm{crossover}}$, defined here as the elapsed time from when the input step is introduced to the onset of initial phase overshoot. Another key metric is $MaxOvershoot$ which indicates the maximum overcorrection in the step response.

Transient simulations seen in Fig. 3 show that $\tau_{\mathrm{crossover}}$ is linearly proportional to the reciprocal of the PLL's -3 dB closed-loop bandwidth: the lower the bandwidth, the higher the $\tau_{\mathrm{crossover}}$. Not surprisingly, $\tau_{\mathrm{crossover}}$ is insensitive to the magnitude of the phase step which is a direct consequence of linearity, a good approximation for small perturbation. In Fig. 3, "52%" and "76%" denote input phase steps equal to 52% and 76% respectively of the reference clock period. Transient simulations also demonstrate, as seen in Fig. 4, that $MaxOvershoot$ is linearly proportional to the maximum gain peaking in the PLL transfer function: the larger the $MaxOvershoot$, the greater the peaking. The magnitude of the overshoot, however, is sensitive to the size of the input

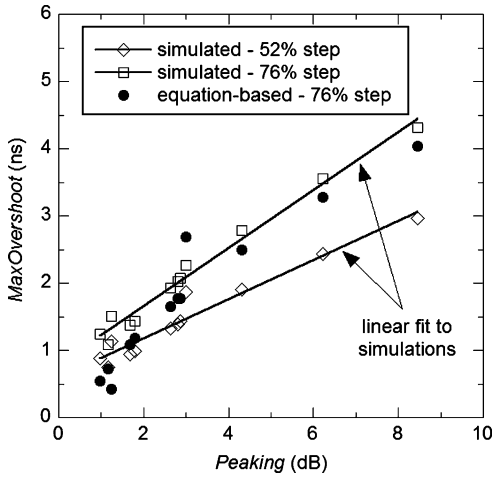


Fig. 4. Simulated and equation-based $MaxOvershoot$ versus peaking for two different input phase steps.

step. Larger steps result in larger $MaxOvershoot$ for a given peaking value, which is another expected consequence of linearity. Bandwidth and peaking values were calculated in Figs. 3 and 4 using continuous-time frequency-domain models of a basic charge-pump PLL [10]. Here, we assumed a charge pump current $(I_{\rm cp})$ feeding a standard loop filter consisting of a series combination of the phase-lead-compensating resistor $(R_{\rm zero})$ and integrating capacitor $(C_{\rm int})$ in parallel with a control voltage ripple smoothing capacitor $(C_{\rm smooth})$ for reducing reference spurs. We also modeled loop delays contributing to phase lag.

To explore the validity of $\tau_{\rm crossover} \propto 1/Bandwidth$ and $MaxOvershoot \propto Peaking$ over a much wider range of loop parameters, we completed additional behavioral simulations to cover low-bandwidth, high-peaking and high-bandwidth, low-peaking scenarios. In these simulations, we varied $R_{\rm zero}$ and $C_{\rm int}$ independently from 0.25× to 4× nominal values, $I_{\rm cp}$ from 0.05× to 20× nominal value, and $C_{\rm smooth}$:$C_{\rm int}$ from 0.01 to 0.1. In the $\tau_{\rm crossover} \propto 1/Bandwidth$ test, the calculated bandwidth was held constant while peaking was swept from 0.25× to 3× nominal value. The result was simulated $\tau_{\rm crossover}$ variation of −22% to +25%. In the $MaxOvershoot \propto Peaking$ test, the simulated peaking was held constant while bandwidth was varied from 0.2× to 4.5× nominal value. The result was simulated $MaxOvershoot$ variation of −12% to +20%. The maximum variations in bandwidth and peaking occurred at the extremes of $R_{\rm zero}$ and $C_{\rm int}$. The relative placement of the loop filter zero and parasitic poles appear to play a role in nonlinear $\tau_{\rm crossover}$ versus $1/Bandwidth$ behavior. In the highest bandwidth cases, low oversampling ratios may also affect the accuracy of the calculated bandwidths and peaking. Despite non-idealities, these simulations still show that $\tau_{\rm crossover} \propto 1/Bandwidth$ and $MaxOvershoot \propto Peaking$ hold across a wide range of loop parameters.

The $\tau_{\rm crossover} \propto 1/Bandwidth$ and $MaxOvershoot \propto Peaking$ relationships can also be deduced analytically to corroborate simulation findings. Approximate closed-form expressions for −3 dB PLL bandwidth and gain peaking already exist for a Type II PLL as functions of damping factor (ζ) and natural frequency (ω_n) [10]:

$$Bandwidth = 2\zeta\omega_n\sqrt{\frac{1}{2}\cdot\left(1+\frac{1}{2\zeta^2}+\sqrt{1+\frac{1}{\zeta^2}+\frac{1}{2\zeta^4}}\right)} \quad (4)$$

$$Peaking = 10\log_{10}\left(\frac{8\zeta^4}{8\zeta^4-4\zeta^2-1+\sqrt{8\zeta^2+1}}\right). \quad (5)$$

In our application, we are specifically interested in how the phase error, $\phi_{\rm err}(t)$, evolves when excited by a phase step, $\phi_{\rm step}$, at $t = 0$. Closed-form expressions for $\phi_{\rm err}(t)$ have been derived in [10] for the cases of critical damping $(\zeta = 1)$, underdamping $(\zeta < 1)$, and overdamping $(\zeta > 1)$: See equations (6)–(8) at the bottom of the page. To gain insight into the relationships between bandwidth and $\tau_{\rm crossover}$, we solve (6)–(8) for $\tau_{\rm crossover}$ using the condition $\phi_{\rm err}(\tau_{\rm crossover}) = 0$. Since $Bandwidth \propto \omega_n$ for a given ζ, it is sufficient to show that $\tau_{\rm crossover} \propto 1/\omega_n$. In principle, we can also obtain $MaxOvershoot$ by solving for $\phi_{\rm err}(t)$ at the first instance $d\phi_{\rm err}/dt$ vanishes to locate the maximum but the mathematics becomes prohibitive.

For a critically damped system, we arrive at

$$\tau_{\rm crossover} = \frac{1}{\omega_n} = \frac{\sqrt{3+2\sqrt{5}}}{Bandwidth} \quad (9)$$

$$MaxOvershoot = \frac{\phi_{\rm step}}{\exp(2)} \quad (10)$$

which clearly illustrate the $\tau_{\rm crossover}$ proportionality to reciprocal bandwidth as well as the $MaxOvershoot$ dependence on $\phi_{\rm step}$ independent of ω_n.

For an underdamped system, we obtain

$$\zeta = 1 : \phi_{\rm err}(t) = \phi_{\rm step}\cdot\exp(-\zeta\omega_n t)\cdot(1-\omega_n t) \quad (6)$$

$$\zeta < 1 : \phi_{\rm err}(t) = \phi_{\rm step}\cdot\exp(-\zeta\omega_n t)\cdot\left[\cos\left(\omega_n t\sqrt{1-\zeta^2}\right) - \frac{\zeta}{\sqrt{1-\zeta^2}}\sin\left(\omega_n t\sqrt{1-\zeta^2}\right)\right] \quad (7)$$

$$\zeta > 1 : \phi_{\rm err}(t) = \phi_{\rm step}\cdot\exp(-\zeta\omega_n t)\cdot\left[\cosh\left(\omega_n t\sqrt{\zeta^2-1}\right) - \frac{\zeta}{\sqrt{1-\zeta^2}}\sinh\left(\omega_n t\sqrt{\zeta^2-1}\right)\right] \quad (8)$$

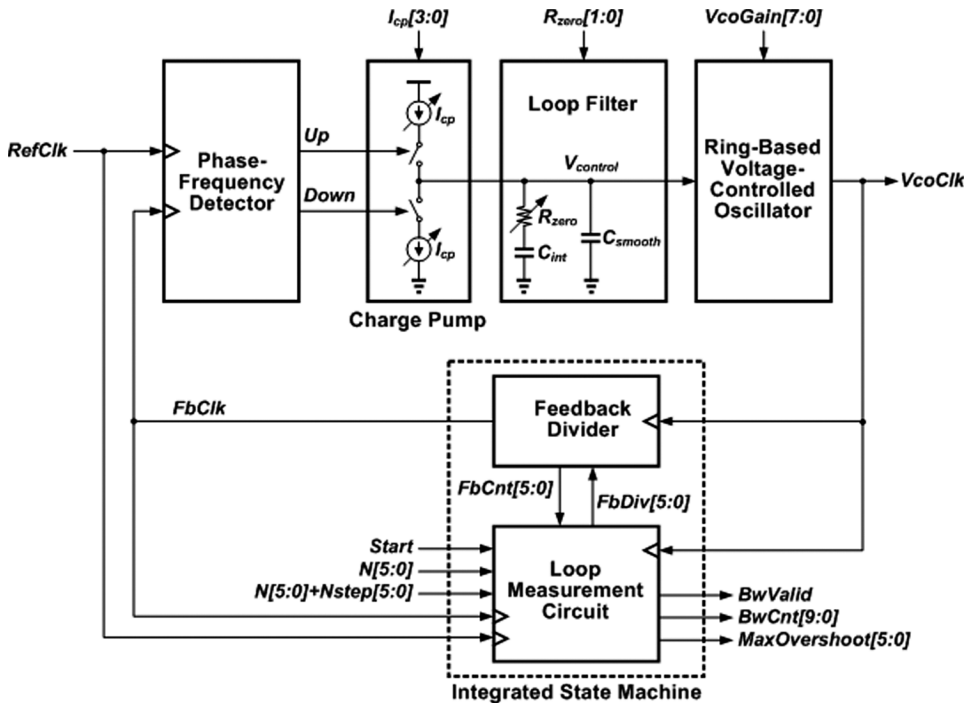


Fig. 5. PLL block diagram with loop measurement circuit integrated in the feedback divider.

$$\tau_{\text{crossover}} = \frac{1}{\omega_n\sqrt{1-\zeta^2}}\arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right) \qquad (11)$$

and once again demonstrate the $\tau_{\text{crossover}}$ to $1/\omega_n$ proportionality. In the limit $\zeta \rightarrow 0$, (4) and (11) can be used to show that $\tau_{\text{crossover}} \rightarrow (\pi/2)/2\omega_n \rightarrow (\pi/2)\sqrt{1+\sqrt{2}}/Bandwidth$.

For an overdamped system:

$$\tau_{\text{crossover}} = \frac{1}{\omega_n\sqrt{\zeta^2-1}}\arctan h\left(\frac{\sqrt{\zeta^2-1}}{\zeta}\right). \qquad (12)$$

At extremely high damping factors ($\zeta \gg 1$), $\tau_{\text{crossover}}$ is proportional to $1/\zeta\omega_n$ while (4) shows that bandwidth approaches $2\zeta\omega_n$. Again, $\tau_{\text{crossover}}$ is proportional to reciprocal bandwidth.

The preceding closed-form equations for phase step response significantly under-predict $MaxOvershoot$ at high damping factors. In order to simplify the mathematics, they assume a loop filter consisting of only R_{zero} and C_{int} and ignore the effect of the parasitic pole introduced by C_{smooth}. The equations also assume a continuous-time system to facilitate a simpler s-domain analysis, an assumption that breaks down from discrete-time aliasing effects [11] as the bandwidth significantly exceeds 10% of the reference clock frequency.

Simulations and closed-form equations show that the relationships between time- and frequency-domain PLL behaviors justify making quick time-domain measurements and then relating the results back to frequency-domain performance specifications. The circuit implementation presented in this paper shows that the PLL step response may be captured by an all-digital, on-chip finite state machine, allowing for fast PLL characterization. Silicon results demonstrate that this circuit can be used for power-on calibration of PLL bandwidth and peaking to compensate for process variations.

III. Implementation of Measurement Circuit

The PLL under test (Fig. 5) is a standard integer-N charge-pump PLL [12]. The only modification is the addition of loop measurement circuitry in the feedback divider state machine. The feedback divider is an incrementing counter clocked by the VCO output. The feedback divisor $(FbDiv[5{:}0])$ is programmable from 5 to 63 although only divisors greater than 7 are used during loop measurement tests. For example, if the feedback divisor is 8, then the counter increments from 0 to 7 before cycling back to 0 and repeating. I_{cp}, R_{zero}, and K_{VCO} are programmable for bandwidth and peaking adjustment as well as for jitter reduction. These adjustments enable a PLL bandwidth of 3 to 25 MHz and peaking of less than 1 to greater than 4 dB to be selected. The ring-based VCO generates 1.6 to 5.0 GHz using a reference input of 100 to 200 MHz. The measurement circuit is exclusively a digital implementation using only standard CMOS library cells to facilitate easier porting to new technology nodes [13]. It interfaces to the existing PLL only at the feedback divider. A standard JTAG scan interface is used to initiate the measurement and retrieve results.

A simple way to generate a reference phase step in a locked PLL is to flip the polarity of the reference clock $(RefClkIn)$ to advance its phase $(RefClkOut)$ by precisely half a clock cycle [14]. One disadvantage of this approach is that the phase step magnitude is only half of the reference clock period. Since $MaxOvershoot$ increases with step size, more accurate measurements can be obtained with larger phase steps. Another disadvantage is that the magnitude of the step is sensitive to reference clock duty cycle distortion (DCD) which may be unknown. Since $MaxOvershoot$ is proportional to the magnitude of the

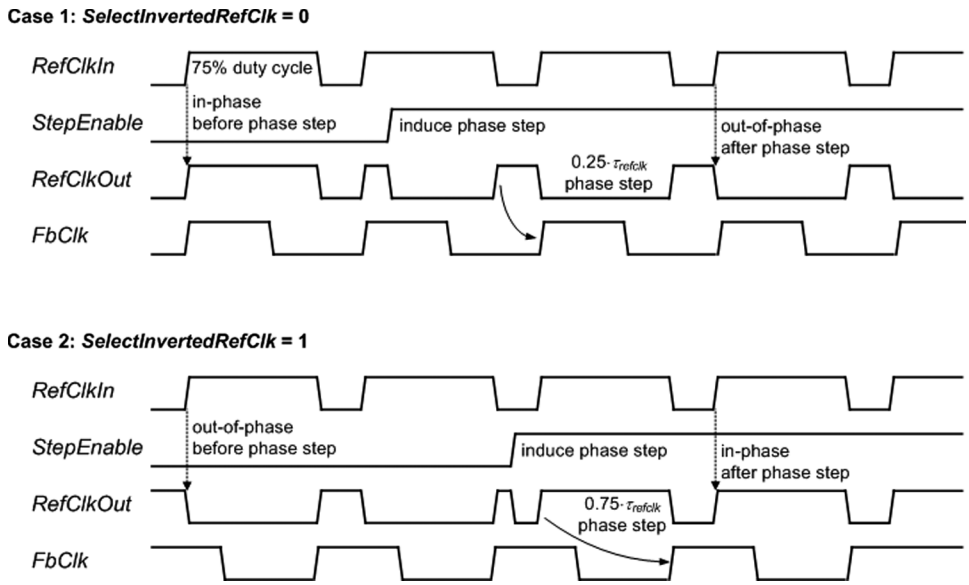


Fig. 6. Timing diagram of reference clock phase step where τ_{refclk} is the reference clock period.

phase step, another approach is necessary. One such approach, illustrated in Fig. 6, is to measure $MaxOvershoot$ twice and then average the results. In the first measurement, the default reference clock polarity is used. In the second, the reference clock polarity is inverted (by asserting a *SelectInvertedRefClk* control signal), the PLL is allowed to re-lock, and then the phase step is introduced. In this way, the average induced phase step over the two measurements is always 50% of the reference clock period regardless of DCD.

With respect to loop dynamics, manipulating the feedback clock phase to introduce a phase step is mathematically equivalent to manipulating the reference clock phase. In this implementation, we manipulate the feedback clock to circumvent DCD concerns and facilitate phase steps as large as the entire reference clock period. The standard loop measurement test consists of three steps. First, the feedback divider is manipulated to introduce a programmable and predictable phase step in the feedback clock. Second, the circuit measures the resulting $\tau_{\mathrm{crossover}}$. Finally, the circuit measures $MaxOvershoot$. Depicted in Fig. 7, the loop measurement circuit consists of three corresponding units: step control, $\tau_{\mathrm{crossover}}$ detector, and $MaxOvershoot$ detector. The step control unit performs the additional function of synchronizing the *Start* signal (*StartRise*) and the reference clock (*RefClk*) into the VCO clock (*VcoClk*) domain as well as generating the *RefRise* and *RefFall* signals that control data flow between *RefClk* and *VcoClk* domains to overcome metastability concerns.

Fig. 8 explains how the input phase step is generated. In this example, the PLL is initially locked to align rising reference and feedback clock edges to each other. A feedback divisor $(FbDiv[5{:}0])$ value of 8 is selected. When the step control unit asserts *StepEn*, $(FbDiv[5{:}0])$ of 11 is loaded into the incrementing feedback divider. This momentary increase in the feedback divisor delays the next rising feedback clock (*FbClk*) by exactly three *VcoClk* cycles to introduce an instantaneous phase error, as is evident in the rising edges of the reference and feedback clocks. The feedback divisor is updated only at a rising feedback clock to avoid corrupting the feedback divider. At the first rising edge of *RefClk* after *StepEn* asserts, the PLL begins to react to the induced phase error by increasing the VCO frequency to re-align the reference and feedback clocks. At the same time (when *RefRise* is asserted), *BwEn* is asserted to enable the counter in the $\tau_{\mathrm{crossover}}$ detector. One feedback clock cycle after *StepEn* is asserted, it is de-asserted, resetting $(FbDiv[5{:}0])$ to its default value of 8. Modifying $(FbDiv[5{:}0])$ for only one feedback clock cycle ensures that the long-term PLL frequency does not drift even as the phase step is applied. If $(FbDiv[5{:}0])$ were otherwise set to 11 indefinitely, the result would be a frequency step applied to the PLL, not just a phase step.

The $\tau_{\mathrm{crossover}}$ detector (Fig. 9) detects the condition when the PLL feedback clock has finally eliminated the induced phase step and begins to lead the reference clock. This marks the onset of phase overshoot and signals the end of the $\tau_{\mathrm{crossover}}$ measurement. The bang-bang (or early-late) phase detector $Q0$ samples the state of the *FbClk* signal at every rising *RefClk*. If *RefClk* leads *FbClk*, then $Q0 = 0$; if *FbClk* leads *RefClk*, then $Q0 = 1$. $Q0$ changing from 0 to 1 indicates the onset of phase overshoot. About two VCO cycles after flip-flop $Q0$ samples the *FbClk* signal, the value of flip-flop $Q0$ is transferred to flip-flop $Q1$, which is clocked in the *VcoClk* domain. At the same time, the previous value of flip-flop $Q1$ is transferred to flip-flop $Q2$. So, flip-flop $Q1$ effectively contains the current value of $Q0$ while flip-flop $Q2$ holds the previous value of $Q0$. If $Q1 = 1$ and $Q2 = 0$, *BwValid* is asserted to end the $\tau_{\mathrm{crossover}}$ test by freezing the $\tau_{\mathrm{crossover}}$ counter $BwCnt[9{:}0]$. The $Q0$-to-$Q1$ transfer is intended solely to reduce the metastability risk during the transfer of data from *RefClk* to *VcoClk* domain.

The 10-bit binary result of the $\tau_{\mathrm{crossover}}$ test $(BwCnt[9{:}0])$ is converted to time using

$$\tau_{\mathrm{crossover}} = \tau_{\mathrm{vcoclk}} \times (BwCnt - N_{\mathrm{step}}) \qquad (13)$$

where τ_{vcoclk} is the nominal *VcoClk* period and N_{step} is the induced phase step size in *VcoClk* cycles. In this example, N_{step}

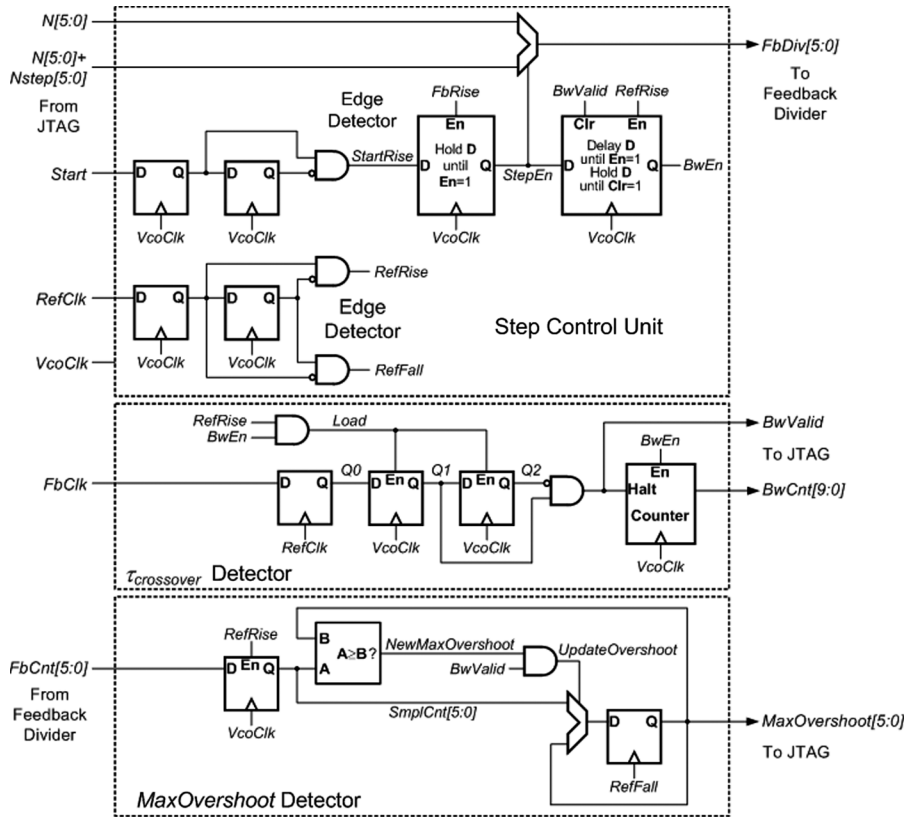


Fig. 7. Block diagram of PLL loop measurement circuits.

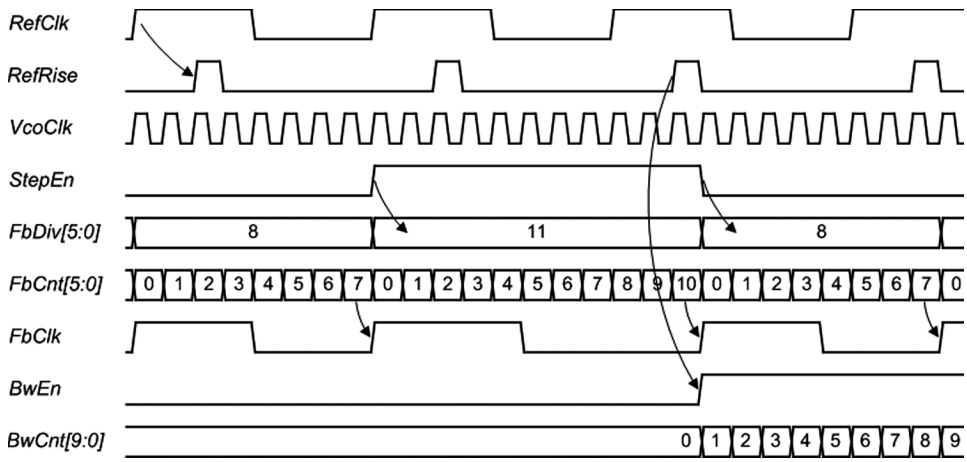


Fig. 8. Timing diagram of phase step induced by the feedback divider.

is 3 (11 minus 8) and must be subtracted from the measurement because the phase step causes the PLL to produce N_{step} additional *VcoClk* cycles during the re-lock process. No other correction factors are necessary as the latencies to start and stop the $\tau_{\text{crossover}}$ counter (measured in *VcoClk* cycles) are equal and cancel each other. The resolution of the $\tau_{\text{crossover}}$ measurement is one *RefClk* period (or τ_{refclk}) and so the measurement becomes much less precise as the PLL bandwidth approaches the *RefClk* frequency. This is a potential limiting factor in the use of this algorithm.

Completion of the $\tau_{\text{crossover}}$ measurement triggers the $MaxOvershoot$ measurement to commence. The $MaxOvershoot$ detector samples the PLL phase error at each rising *RefClk* edge, searching for the largest phase overshoot. Rather than using a time-to-digital converter to determine the phase error, the $MaxOvershoot$ detector samples the internal state of the feedback divider at rising *RefClk* edges to provide a digital representation of the instantaneous phase error. The least significant bit of this sampled divider state $(SmplCnt[5{:}0])$ is equivalent to one *VcoClk* period. Fig. 10(a) traces the transient response of $SmplCnt[5{:}0]$ for the high-peaking case of Fig. 2. In this example, when the PLL is locked, $SmplCnt = 0$ and the PLL feedback divisor is set to 25. The reference clock is advanced by 19 *VcoClk* cycles at time $= 0$, resulting in an initial $SmplCnt[5{:}0]$ of 6 (25 minus 19). The phase error is eliminated over time which

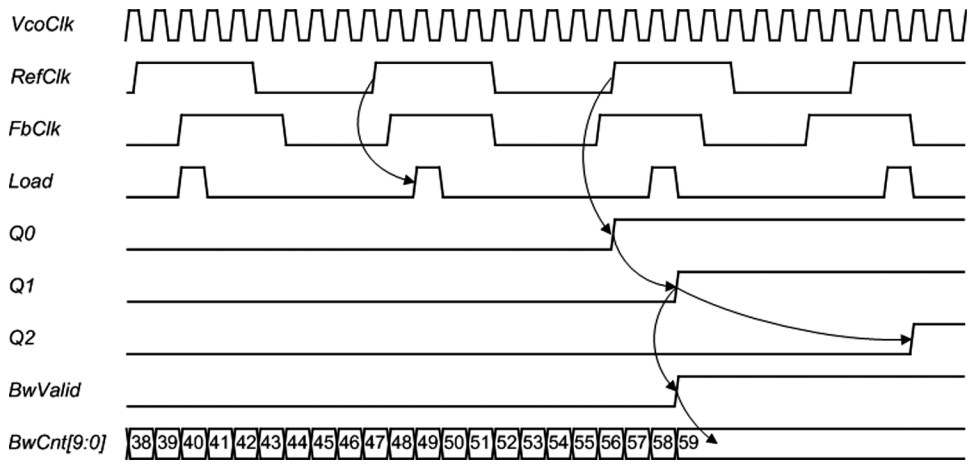


Fig. 9. Timing diagram of $\tau_{\mathrm{crossover}}$ measurement.

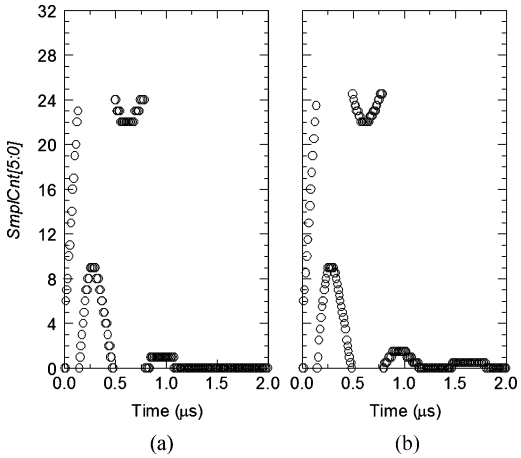


Fig. 10. Simulated PLL transient step response captured from sampling feedback divider state at (a) rising *RefClk* edges only and (b) both rising and falling *RefClk* edges for improved resolution.

causes $SmplCnt[5{:}0]$ to reach a maximum value of 24 and then wraps around zero (at time $= 0.14$ μs) to mark the onset of phase overshoot. During initial phase overshoot, the instantaneous VCO frequency is higher than the nominal frequency and so the feedback clock pulls increasingly ahead of *RefClk*. This results in $SmplCnt[5{:}0]$ values increasing over time, reaching a maximum value of 9 at time $= 0.28$ μs. Eventually, PLL feedback corrects the VCO frequency error and both the resulting phase error and $SmplCnt[5{:}0]$ values begin to decrease. At time $= 0.48$ μs, $SmplCnt[5{:}0]$ wraps back from 0 to 24, marking the onset of undershoot. Maximum undershoot is seen at time $= 0.61$ μs where $SmplCnt[5{:}0] = 22$ suggests a maximum undershoot of 3 (25 minus 22). A second, smaller overshoot of one VCO cycle appears at time $= 0.95$ μs.

Fig. 11 shows the timing diagram for the implemented $MaxOvershoot$ detector (Fig. 7). Note that the feedback divisor in this diagram is set to 8, the same as in Figs. 8 and 9. The detector samples the current internal state of the feedback divider ($FbCnt[5{:}0]$) at every *RefRise* pulse (where *RefRise* is a synchronized version of the rising *RefClk* edge) and captures the result in the $SmplCnt[5{:}0]$ register. In this implementation, the current $SmplCnt[5{:}0]$ is compared to the previous maximum overshoot ($MaxOvershoot[5{:}0]$). If $SmplCnt[5{:}0]$ is greater than the previous maximum value, then $SmplCnt[5{:}0]$ replaces the previous $MaxOvershoot[5{:}0]$ at the next *RefFall* pulse. *RefFall* clocks the data transfer from $SmplCnt[5{:}0]$ to $MaxOvershoot[5{:}0]$ to allow for comparator latency. To filter sampled values associated with phase undershoot, the comparator ignores $SmplCnt[5{:}0]$ values greater than $FbDiv[5{:}0]/2$. In this example with a feedback divisor of 8, $SmplCnt[5{:}0]$ values of 7, 6, and 5 are ignored.

Since the feedback divider state is sampled at the assertion of the synchronized *RefRise* pulse rather than by *RefClk* itself, the measured $MaxOvershoot$ result includes the *RefRise* synchronizer latency (measured in *VcoClk* cycles). This synchronizer latency (K_{sync}) must be subtracted from the measured $MaxOvershoot$ count to calculate the actual maximum overshoot. K_{sync} is captured in a separate test mode where the feedback divider state is sampled (as previously described) but no phase step is applied. Since there is no overshoot to measure, the measured $SmplCnt[5{:}0]$ value in this test mode is the synchronizer latency.

The 6-bit binary result of the $MaxOvershoot$ test, measured in *VcoClk* cycles, is converted to time using

$$\tau_{MaxOvershoot} = \tau_{\mathrm{vcoclk}} \times (MaxOvershoot - K_{\mathrm{sync}}). \quad (14)$$

The resolution of the $MaxOvershoot$ measurement is $\tau_{\mathrm{refclk}}/FbDiv$ (or τ_{vcoclk}), and so the measurement is less precise in PLLs with small feedback divisors as well as minimal peaking due to quantization effects. Fortunately, the $MaxOvershoot$ resolution can be doubled by synchronizing *RefClk* to both true and complement phases of *VcoClk* and adding some logic and flip-flops to determine the *VcoClk* phase in which the rising *RefClk* appears. The resulting improvement in resolution is exemplified in Fig. 10(b). So conceptually, if *P* *VcoClk* phases are available, $MaxOvershoot$ uncertainty can correspondingly be reduced to τ_{vco}/P.

Although we retard the feedback clock phase in our measurements, we can also advance the feedback clock phase by momentarily decreasing $FbDiv[5{:}0]$. If the nominal feedback divisor is close to the maximum value of 63, then advancing

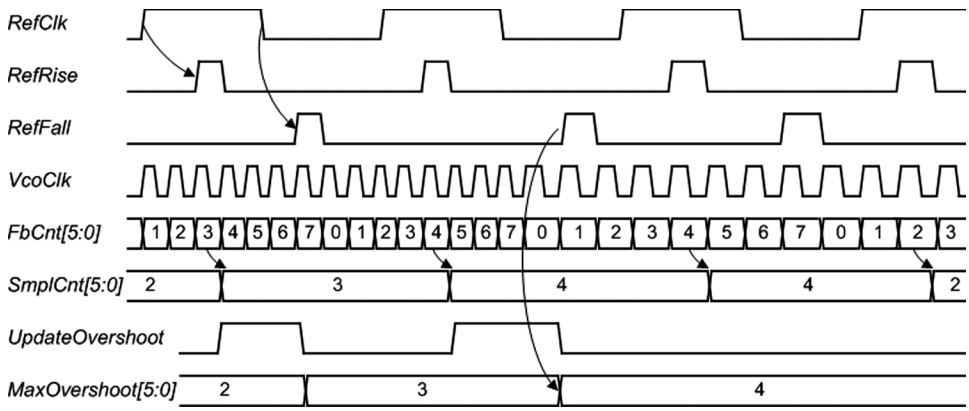


Fig. 11. Timing diagram of $MaxOvershoot$ measurement.

the feedback clock phase allows for a larger phase step. However, one downside of advancing the feedback clock phase is inability to detect phase overshoots less than the *RefClk* synchronizer latency.

If the PLL static phase error is larger than the maximum phase overshoot, then the required change in the phase error polarity does not occur and the $\tau_{\rm crossover}$ measurement does not complete. In this case, the $\tau_{\rm crossover}$ counter saturates at its maximum value and *BwValid* remains low to indicate a failed test. If this occurs, the feedback clock phase can be advanced instead of retarded to guarantee a phase error polarity change. As the static phase error magnitude approaches the maximum phase overshoot, the measurement accuracy is degraded. However, in this PLL, the static phase error is less than 10% of the *VcoClk* period while the expected $MaxOvershoot$ is at least several *VcoClk* periods. Behavioral simulations show that a static phase error as large as 50% of the *VcoClk* period produces negligible impact on measurement accuracy.

The loop measurement circuit can also be used to generate the time evolution of the PLL step response, similar to a time interval error (TIE) plot. Instead of automatically detecting $\tau_{\rm crossover}$ and $MaxOvershoot$, the feedback divider count is captured after exactly *N RefClk* cycles. By varying N from 1 to the maximum value of 63, the transient of the PLL step response may be plotted as in Fig. 10.

The loop measurement circuit may also be used as a lock detector by repeatedly measuring $K_{\rm sync}$. If $K_{\rm sync}$ does not vary, then the PLL is locked. The static phase error may be estimated by comparing the measured $K_{\rm sync}$ to the expected synchronizer latency of two *VcoClk* cycles.

All loop measurement clocks are gated to minimize power consumption when not in use. Sense-amplifier type flip-flops are used for short setup time and quick resolution out of metastability. In this implementation, *VcoClk* clocks most of the state machine. If technology imposes timing constraints at VCO clock frequencies, the state machine can be easily re-designed to be clocked by the slower reference clock as in [14]. In this case, the phase step should be generated by inverting the reference clock as in Fig. 6.

IV. Experimental Results

The presented loop measurement algorithms and circuits have been successfully integrated into a range of 65 nm, 45 nm, and 32 nm processor products over a wide range of operating frequencies [15]–[17]. In this paper, we focus on measurements from two different PLL designs operating at 2.5 GHz with a 100 MHz reference clock input. Both PLLs were fabricated using a high-performance logic 45 nm partially-depleted SOI-CMOS technology [18], [19]. The first PLL (PLL1) was described in Section III. Details of the second PLL (PLL2) are presented in [14]. The loop measurement circuit of PLL2 is only different in that the phase step is introduced by manipulating the reference clock. We include PLL2 measurements to prove that the presented algorithms and circuits are effective across a broad set of loop parameters. Programmable $I_{\rm cp}$ and $R_{\rm zero}$ are used to vary bandwidth and peaking in both PLLs. Nominal $K_{\rm VCO}$ is 10 GHz/V for PLL1 and 2.7 GHz/V for PLL2. PLL2 has additional $K_{\rm VCO}$ programmability of 0.55× to 1.4× nominal value. $C_{\rm int}$ is fixed at 19.9 pF (PLL1) and 40 pF (PLL2) while $C_{\rm smooth}$ is fixed at 1.05 pF (PLL1) and 1.3 pF (PLL2).

Tables I and II compare measured and simulated bandwidth and peaking at various $I_{\rm cp} - R_{\rm zero}$ combinations for PLL1 and PLL2 respectively. We report results from three PLL1 parts and a single PLL2 part. As an illustration of phase error transients captured from this loop measurement circuit, the step responses of Fig. 2 actually correspond to PLL1 Cases 2 and 9 in Table I. For PLL1, the measured results for Cases 10 to 12 are nearly identical, probably due to premature $I_{\rm cp}$ saturation. The unexpectedly similar measurement results for Cases 1 and 2 are likely due to second-order charge-pump effects at small $I_{\rm cp}$. In general, measured bandwidths are higher than simulated values while measured peaking is lower. For PLL1, the loop measurement circuit captured $\tau_{\rm crossover}$ and $MaxOvershoot$ as described in Section II using phase steps of $+13$ and $+19$ *VcoClk* cycles, respectively comprising 52% and 76% of the *RefClk* period. For each step size and PLL setting, the loop measurement test was run 25 times to confirm repeatability. Indeed, run-to-run variation was bounded to within the measurement resolution. For PLL2, $\tau_{\rm crossover}$ and $MaxOvershoot$ results were captured in a similar fashion but a phase step of 50% was applied.

TABLE I
COMPARISON OF SIMULATED AND MEASURED LOOP PARAMETERS AT VARIOUS PLL1 SETTINGS

Case	R_{zero} (kΩ)	I_{cp} (μA)	*Bandwidth* (MHz)				*Peaking* (dB)			
			Simulated	Measured			Simulated	Measured		
				Part 1	Part 2	Part 3		Part 1	Part 2	Part 3
1	3.2	2.5	1.8	3.4	3.4	3.0	8.5	4.2	4.1	4.2
2	3.2	5	2.6	3.1	3.2	3.2	6.2	4.3	4.0	4.2
3	3.2	10	4.0	5.0	5.4	5.3	4.3	3.0	2.8	2.8
4	3.2	20	6.6	9.3	10.0	10.0	2.9	1.8	1.5	1.7
5	4.8	10	4.8	6.2	6.8	6.5	2.6	1.7	1.5	1.3
6	4.8	20	9.0	13.2	14.8	14.2	1.7	0.9	0.7	0.7
7	6.4	5	3.3	4.3	4.5	4.5	2.8	1.9	2.0	1.7
8	6.4	10	6.0	8.1	9.1	9.0	1.8	1.1	1.0	1.0
9	6.4	20	12.0	17.6	19.7	18.7	1.2	0.7	1.1	1.0
10	6.4	30	18.1	25.6	27.1	26.2	0.8	2.1	2.6	2.8
11	6.4	40	23.3	25.7	26.8	26.1	1.2	2.2	2.5	2.8
12	6.4	70	35.2	25.7	26.7	25.9	3.0	2.1	2.3	2.6

TABLE II
COMPARISON OF SIMULATED AND MEASURED LOOP PARAMETERS AT VARIOUS PLL2 SETTINGS

Case	K_{VCO} (GHz/V)	R_{zero} (kΩ)	I_{cp} (μA)	*Bandwidth* (MHz)		*Peaking* (dB)	
				Simulated	Measured Part 1	Simulated	Measured Part 1
1	3.8	7	86	10.7	9.2	0.65	0.23
2	3.8	6	102	10.9	9.1	0.71	0.20
3	3.5	6	102	10.0	8.1	0.75	0.26
4	3.0	6	102	8.2	7.1	0.84	0.36
5	2.7	6	102	7.5	6.6	0.89	0.41
6	2.7	7	102	9.0	7.9	0.72	0.20
7	2.7	8	102	10.4	9.2	0.61	0.19
8	2.7	10	102	12.4	10.6	0.98	0.87
9	2.7	7	86	7.5	6.7	0.81	0.30
10	2.7	8	73	7.3	7.0	0.75	0.19
11	2.7	10	44	5.4	7.1	0.79	0.16
12	2.7	6	25	2.0	1.6	2.49	1.81
13	2.7	10	25	3.0	2.4	1.20	0.59
14	2.7	10	44	5.4	4.7	0.79	0.27
15	2.7	10	86	10.7	9.5	0.69	0.56
16	2.7	10	125	14.6	12.1	1.50	1.41
17	2.7	10	175	18.5	15.2	2.67	2.46
18	2.7	6	44	3.2	2.7	1.66	1.05
19	2.7	6	62	4.4	4.0	1.29	0.67
20	2.7	6	86	6.3	5.6	1.01	0.47
21	2.7	6	125	9.5	7.9	0.77	0.27
22	2.7	6	175	13.6	11.3	0.63	0.17
23	2.7	8	86	8.7	8.0	0.67	0.13
24	2.7	7	125	11.2	9.4	0.63	0.20
25	2.7	8	125	12.7	10.7	0.64	0.35
26	2.5	6	102	6.9	6.1	0.95	0.51
27	2.0	6	102	5.5	4.9	1.11	0.53
28	1.5	6	102	3.9	3.1	1.42	0.86

The measured $\tau_{\text{crossover}}$ (Fig. 12) shows the same linear relationship to reciprocal bandwidth as simulated data. The slopes of the linear fits to measured data are the same for both PLLs and are about 10% larger than the slope of the linear fit to simulated data. The linear fit y-intercept for PLL1 is approximately one *RefClk* period (10 ns) higher than for the simulated data while the y-intercept for PLL2 is approximately a half *RefClk* period (5 ns) higher than for simulated data. $\tau_{\text{crossover}}$ measurements are prone to quantization error effects described in Section III. Since this error is always positive, we expect that the average measured $\tau_{\text{crossover}}$ will somewhat exceed the simulated $\tau_{\text{crossover}}$ as the latter contains no quantization error. For a few PLL1 settings, the measured $\tau_{\text{crossover}}$ is slightly higher with the 76% step than with the 52% step although the differences do not exceed the temporal resolution of the test (τ_{refclk}).

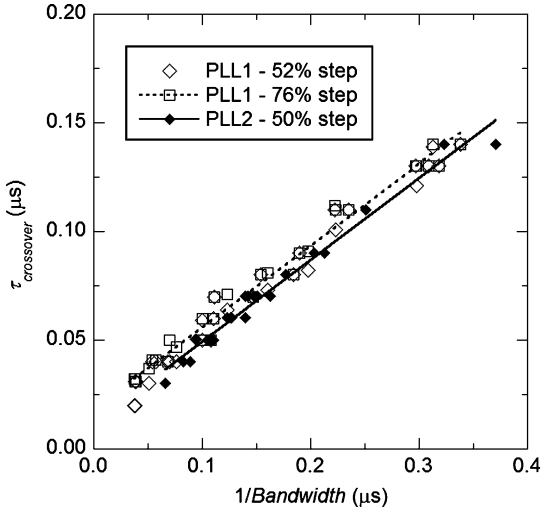


Fig. 12. Measured $\tau_{\text{crossover}}$ versus reciprocal bandwidth.

For both PLLs, the linear fit coefficients for $\tau_{\text{crossover}}$ to $1/Bandwidth$ were used to estimate bandwidth from measured $\tau_{\text{crossover}}$ values. Fig. 13 shows the errors in bandwidth estimated from $\tau_{\text{crossover}}$ versus measured bandwidth obtained from reference spur bench measurements. Errors are smallest in the cases of low PLL bandwidth as the quantization errors are correspondingly small. For bandwidths lower than 9 MHz, the errors fall within 1 MHz. For bandwidths of 9 to 20 MHz, the errors are less than 3 MHz. The errors are less than 4 MHz for bandwidths of 20 to 30 MHz. In all cases, the predicted bandwidth errors are within the bounds predicted by quantization effects.

Fig. 14 shows the relationship between measured $MaxOvershoot$ and measured gain peaking. Although the measurement results exhibit significant quantization effects, the slopes of the linear fits for both PLLs closely match the slopes of the simulated data, supporting the premise that $MaxOvershoot$ is proportional to peaking. The slope of the linear fit in the PLL1 76% phase step case is 43% higher than in the 52% phase step case, close to the expected increase of 46%. The $MaxOvershoot$ measurements follow the linear fits within the measurement resolution (τ_{vcoclk}) with only one exception.

For both PLLs, the linear fit of $MaxOvershoot$ to gain peaking was used to estimate peaking from measured $MaxOvershoot$ values. Fig. 15 shows the errors in peaking estimated from $MaxOvershoot$ versus measured peaking

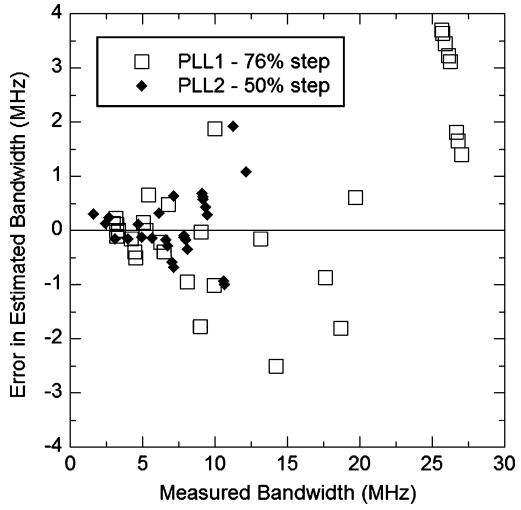


Fig. 13. Errors in estimated bandwidth from $\tau_{\mathrm{crossover}}$ versus measured bandwidth.

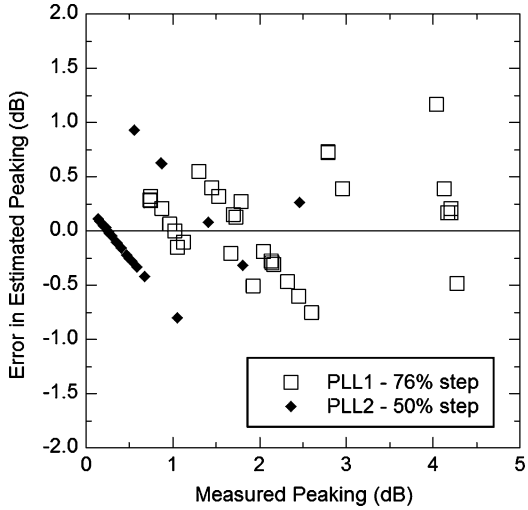


Fig. 15. Errors in estimated peaking from $MaxOvershoot$ versus measured peaking.

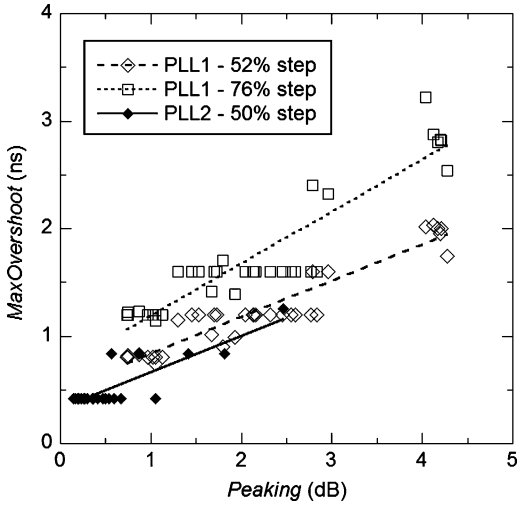


Fig. 14. Measured $MaxOvershoot$ versus peaking.

obtained from reference spur bench measurements. The errors ranged from -1.0 to $+1.2$ dB. The repeating, linear patterns running diagonally from top-left to bottom-right for both PLLs show strong quantization effects. Such effects strongly motivate the need for increased resolution in the $MaxOvershoot$ detector as described in Section III. For example, simply sampling the reference clock with both edges of the VCO clock should halve these errors. Note that although peaking in Figs. 14 and 15 is plotted in dB, $MaxOvershoot$ is more accurately related to peaking plotted on a linear scale. However, for peaking values of 0.5 to 6.0 dB, the results remain nearly unchanged when peaking is plotted on a linear scale since $\log(x)$ is almost linearly related to x in this peaking range.

The bandwidth and peaking errors in Figs. 13 and 15, respectively, were analyzed to assess the effectiveness of the loop measurement circuit in identifying passing and failing parts. For PLL1, the pass criterion was the PCI Express 2.0 specification -8 to 16 MHz bandwidth with peaking below 3 dB. The loop measurement circuit correctly identified all 27 measurements with an out-of-specification bandwidth. The circuit did misclassify two of nine passing cases, one of which was 1.2 MHz within specification. The circuit correctly identified all six cases with peaking that was out of specification. However, three of 30 passing cases were misclassified as failing. In the worst case, the misclassified setting passed the specification by 0.21 dB. For PLL2, the applicable pass criterion was the alternative PCI Express 2.0 specification -5 to 8 MHz bandwidth with peaking below 1 dB. The loop measurement circuit correctly identified 17 of 18 settings that produced an out-of-specification bandwidth. In the one failing case that was missed, the measured bandwidth was only 0.1 MHz outside the passing range. The circuit correctly classified all ten passing settings. The circuit failed to identify one of four settings that failed the peaking requirement. In the missed case, the actual peaking was 0.05 dB outside the specification. However, three of 24 passing cases were misidentified as failing. In the worst case, the misclassified setting passed the specification by 0.44 dB.

The simulated power consumption for the loop measurement circuit in both PLL1 and PLL2 is about 2.5 mW when operating at 2.5 GHz on a 1.2-V power supply. The silicon area is 2750 μm^2, although it can easily be reduced by 40–50% by replacing some non-critical sense-amplifier flip-flops with smaller master-slave flip-flops and by optimizing the overshoot comparator. Layout area was not a serious constraint in this design. The die micrograph of a 45 nm processor product with PLL1 is shown in Fig. 16 [20]. Fig. 17 highlights the relative size and location of the loop measurement circuit with respect to the floorplan and micrograph of PLL2 [14].

V. Conclusion

An on-chip, all-digital state machine can be used to accurately estimate PLL bandwidth and peaking with potentially large savings in tester time. This design-for-test feature may be used from wafer- to package-level testing, minimizing die and package waste and allowing for adaptive PLL loop calibration.

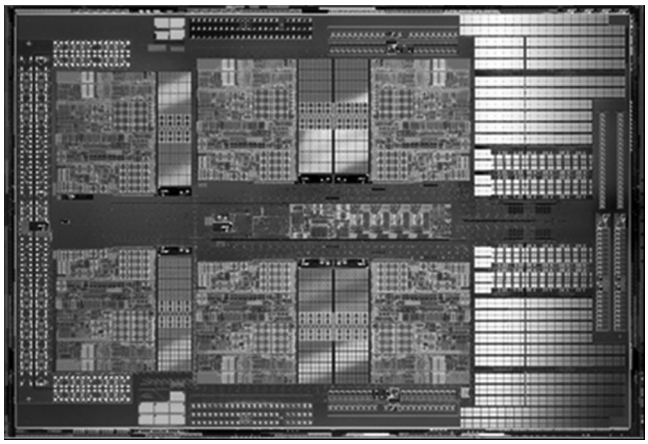


Fig. 16. Die micrograph of exemplary 45 nm processor product with loop measurement circuit for PLL1 [20].

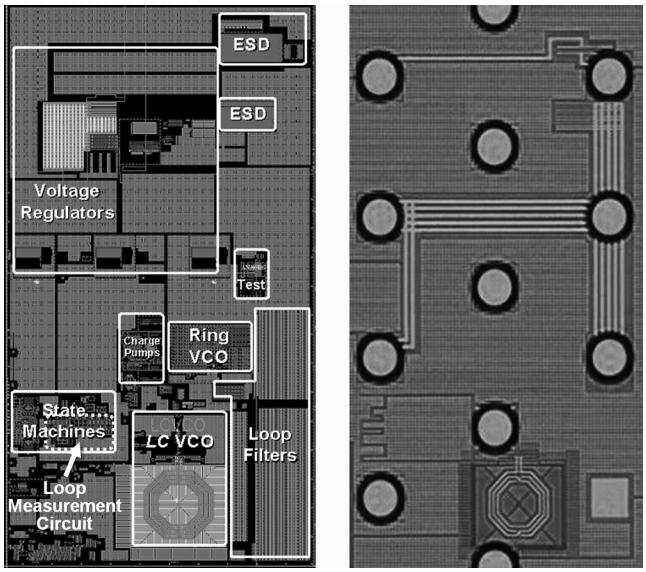


Fig. 17. PLL2 die floorplan (left) and microphotograph (right) highlighting PLL loop measurement circuit [14].

ACKNOWLEDGMENT

The authors thank John Lee (now at the Massachusetts Institute of Technology) and Anand Thiruvengadam, and acknowledge AMD management from Geoff Brehmer, Dru Cabler, Bruce Doyle, Emerson Fang, and Mike Leary for supporting the development of this work.

REFERENCES

[1] *PCI Express 2.0*, [Online]. Available: http://www.pcisig.com/specifications/pciexpress/specifications

[2] R. C. Dixon, *Spread Spectrum Systems With Commercial Applications*, 3rd ed. New York: Wiley-Interscience, 1994.

[3] M. Li, A. Martwick, G. Talbot, and J. Wilstrup, "Transfer functions for the reference clock jitter in a serial link: Theory and applications," in *Proc. IEEE Int. Test Conf.*, Charlotte, NC, Oct. 2004, pp. 1158–1167.

[4] A. L. S. Loke, B. A. Doyle, M. M. Oshima, W. L. Williams, R. G. Lewis, C. L. Wang, A. Hanpachern, K. M. Tucker, P. Gurunath, G. C. Asada, C. O. Lackey, T. T. Wee, and E. S. Fang, "Loopback architecture for wafer-level at-speed testing of embedded HyperTransport™ processor links," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, CA, Sep. 2009, pp. 605–608.

[5] S. Kim and M. Soma, "An all-digital built-in self-test for high-speed phase-locked loops," *IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process.*, vol. 48, no. 2, pp. 141–150, Feb. 2001.

[6] B. R. Veillette and G. W. Roberts, "On-chip measurement of the jitter transfer function of charge-pump phase-locked loops," *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 483–491, Mar. 1998.

[7] S. Sunter and A. Roy, "BIST for phase-locked loops in digital applications," in *Proc. IEEE Int. Test Conf.*, Atlantic City, NJ, 1999, pp. 532–540.

[8] Y. Akamine, M. Kawabe, K. Hori, T. Okazaki, M. Kasahara, and S. Tanaka, "$\Delta\Sigma$ transmitter with a loop-bandwidth calibration system," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 497–506, Feb. 2008.

[9] D. M. Fischette, R. M. Desantis, and J. H. Lee, "An on-chip all-digital measurement circuit to characterize PLL loop response in 45 nm SOI," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, CA, Sep. 2009, pp. 609–612.

[10] F. M. Gardner, *Phaselock Techniques*, 3rd ed. New York: Wiley, 2005.

[11] A. L. S. Loke, R. K. Barnes, T. T. Wee, M. M. Oshima, C. E. Moore, R. R. Kennedy, and M. J. Gilsdorf, "A versatile 90-nm CMOS charge-pump PLL for SerDes transmitter clocking," *IEEE J. Solid-State Circuits*, vol. 41, no. 8, pp. 1894–1907, Aug. 2006.

[12] F. M. Gardner, "Charge-pump phase-locked loops," *IEEE Trans. Commun.*, vol. COM-28, no. 11, pp. 1849–1858, Nov. 1980.

[13] L. Bair, "Process/product interactions in a concurrent design environment," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, CA, Sep. 2007, pp. 779–782.

[14] D. M. Fischette, A. L. S. Loke, M. M. Oshima, B. A. Doyle, R. Bakalski, R. J. DeSantis, A. Thiruvengadam, C. L. Wang, G. R. Talbot, and E. S. Fang, "A 45 nm SOI-CMOS dual-PLL processor clock system for multi-protocol I/O," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2010, pp. 246–247.

[15] M. Golden, S. Arekapudi, G. Dabney, M. Haertel, S. Hale, L. Herlinger, Y. Kim, K. McGrath, V. Palisetti, and M. Singh, "A 2.6 GHz dual-core 64b $\times$86 microprocessor with DDR2 memory support," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2006, pp. 128–129.

[16] J. Dorsey, S. Searles, M. Ciraula, S. Johnson, N. Bujanos, D. Wu, M. Braganza, S. Meyers, E. Fang, and R. Kumar, "An integrated quad-core opteron™ processor," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2007, pp. 102–103.

[17] R. Jotwani, S. Sundaram, S. Kosonocky, A. Schaefer, V. Andrade, G. Constant, A. Novak, and S. Naffziger, "An x86–64 core implemented in 32 nm SOI CMOS," in *IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers*, San Francisco, CA, Feb. 2010, pp. 106–107.

[18] S. Narasimha *et al.*, "High-performance 45-nm SOI technology with enhanced strain, porous low-k BEOL, and immersion lithography," in *IEEE Int. Electron Devices Meeting Tech. Dig.*, San Francisco, CA, Dec. 2006.

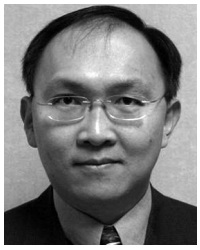
[19] M. Horstmann, A. Wei, J. Hoentschel, T. Feudel, T. Scheiper, R. Stephan, M. Gerhadt, S. Krügel, and M. Raab, "Advanced SOI CMOS transistor technologies for high-performance microprocessor applications," in *Proc. IEEE Custom Integrated Circuits Conf.*, San Jose, CA, Sep. 2009, pp. 149–152.

[20] P. Conway, N. Kalyanasundharam, G. Donley, K. Lepak, and B. Hughes, "Blade computing with the AMD Opteron™ processor," in *IEEE Hot Chips*, Stanford, CA, 2009.

Dennis M. Fischette (M'95) received the B.S. degree in applied and engineering physics in 1986 from Cornell University, Ithaca, NY, and studied the History of Science at the University of California, Berkeley, from 1986 to 1988.

He is a Principal Member of Technical Staff at Advanced Micro Devices (AMD), Sunnyvale, CA. His technical interests include PLL and high-speed I/O design and circuit analysis software. Before joining AMD, he worked for Integrated CMOS Systems, HaL Computer Systems, and Chromatic Research as a custom circuit designer (analog, digital, and memory) and CAD software developer. He is the author of three technical publications and has taught invited short courses at ISSCC and Symposia on VLSI Technology and Circuits.

Mr. Fischette is a former IEEE Solid-State Circuits Society Distinguished Lecturer and ISSCC technical committee member and currently serves as a CICC technical program committee member.



Alvin L. S. Loke (S'89–M'99–SM'04) received the B.A.Sc. degree in engineering physics with highest honors from the University of British Columbia, Vancouver, Canada, in 1992, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1999, respectively. He was a recipient of the Canadian NSERC 1967 Graduate Scholarship and his doctoral research focused on interconnect integration and reliability of copper and low-K dielectrics.

He has interned at Sumitomo Electric Industries (Osaka, Japan), Texas Instruments (Dallas, TX), and Motorola (Austin, TX). After graduating, he worked for several years on technology integration at Hewlett-Packard Laboratories (Palo Alto, CA) and at Chartered Semiconductor Manufacturing (Singapore) as an Agilent (now Avago) Technologies assignee. He later transferred to Fort Collins, CO, to design SerDes PLL/DLL circuits. In 2006, he joined Advanced Micro Devices where he is a Senior Member of Technical Staff designing wireline circuits and architectures as well as interfacing with technology groups on analog/mixed-signal concerns. He has authored over 30 technical publications and holds ten patents.

Since 2003, Dr. Loke has chaired and remains active in the Fort Collins Solid-State Circuits Society (SSCS) Technical Chapter which received the Outstanding Chapter Award in 2005. He has been on the CICC technical program committee since 2006 and serves on the ECE Department Industrial Advisory Board of Colorado State University (recently as President) and the SSCS Chapters committee.

Richard J. DeSantis received the B.S. degree in electrical engineering from the Rochester Institute of Technology, Rochester, NY, in 1979.

In 1979, he joined International Business Machines, Endicott, NY, where he worked on test development/manufacturing for midrange processors, line impact printers, infrared laser optoelectronics for 1-Gb/s SerDes channel interconnects, and hard disk drive arm electronics. In 1994, he joined HaL Computers, Campbell, CA, where he was involved in laboratory test development of ASICs used in high-speed parallel interconnects for Intel's Coherent and Clustered multiprocessor servers. In 2002, he joined Advanced Micro Devices (AMD), Sunnyvale, CA. He reported to the AMD Opteron™'s (K8) system architect focusing on laboratory test characterization for HyperTransport I/O and PLLs used in AMD processors. Currently a Member of the Technical Staff, he works in the Analog Mixed Signal Center of Excellence group that is responsible for AMD processor's HyperTransport™ and PLL development. He continues to be responsible for laboratory test characterization.

Gerry R. Talbot (M'02) received the B.Sc. degree in electrical and electronic engineering from Portsmouth University, U.K., in 1979 and started work on microprocessor design and serial interconnects for Inmos.

He is a Senior Fellow at Advanced Micro Devices (AMD), Boxborough, MA, where he has worked since 2002. His primary focus is in high speed I/O and memory interconnects, he is involved in the development of, and contributing to industry standard specifications such as HyperTransport™ and PCI Express®. His work involves silicon circuit design, system-level jitter modeling, system channel modeling, and device measurement. Before joining AMD, he worked for several computer system companies designing and developing a range of computing systems from massively parallel supercomputers to rack-mounted servers. His main contributions throughout all of these projects have been in the areas of computer architecture, system and silicon design and high-speed interconnect development. He holds 34 patents in computer system and mixed-signal circuit design.