A Fast Methodology for First-Time-Correct Design of PLLs using Nonlinear Phase-Domain VCO Macromodels

Prashant Goyal
Indian Institute of Technology, Kanpur, India
Xiaolue Lai, Jaijeet Roychowdhury
University of Minnesota, Twin Cities, USA

Abstract— **We present a novel methodology suitable for fast, correct design of modern PLLs. The central feature of the methodology is its use of accurate, nonlinear behavioral models for the VCO within the PLL, thus removing the need for many time-consuming SPICE-level simulations during the design process. We apply the new methodology to design a novel injection-aided PLL that acquires lock 3× faster than prior designs, without trading off other design metrics such as jitter. We demonstrate how existing design methodologies based on behavioral simulation are incapable of leading to our new PLL design. The nonlinear behavioral simulations employed in our methodology are about 2 orders of magnitude faster than transistor-level ones, resulting in an overall design productivity gain of an order of magnitude.**

I. Introduction

Phase locked loops or PLLs are important in virtually all mixed-signal and digital systems. For example, PLL synthesizers are employed frequently in mobile communications and wireless communication transceivers. In high-speed data communications systems such as Ethernet transceivers, disk drive read/write channels, digital mobile receivers, high-speed memory interfaces and so forth, PLLs are widely used as clock generators. Other uses include clock and data recovery (CDR) direct FM-demodulation in RF systems (see, *eg.*, [1]).

The design of PLLs constitutes one of the most challenging problems in mixed-signal design today. Because of complex nonlinear dynamics in their transient operation, achieving the right balance between various PLL design metrics — such as settling time, phase noise or jitter performance, lock and capture ranges, *etc.*— for a given application is far from simple. It is not uncommon, therefore, for many months to be required to finalize the design of today's advanced PLLs. Employing effective design methodologies, supported heavily by simulation at different abstraction levels, is crucial in PLL design. Unfortunately, existing methodologies for PLL design are often inefficient or ineffective, with the result that it is not uncommon for 5 or more re-spins to be required before the PLL functions correctly.

In existing methodologies, a fresh PLL design often starts from a simple first-principles block structure such as that shown in Figure 1, or from an existing PLL design. Rough hand calculations, based on simple classical linearized analysis of a PLL feedback loop in lock, are first performed by the designer to estimate lock range, jitter, *etc.*. During the course of the design, behavioral simulation using *phase-domain macromodels* is extensively applied for greater accuracy. When the design is finalized at the transistor level, full SPICE-level simulation is heavily used for final verification.

Important steps in this flow break down in today's methodologies. It is for this reason that, as mentioned above, PLL design tends to be extremely time consuming and error prone. Problems exist at each level of the above flow that contribute to the breakdown:

• hand calculation level: Existing hand-analysis techniques [1], [2] for dynamics, noise, jitter, *etc.*, in PLLs are all based on linear analysis of the PLL around a locked steady state. The few nonlinear analyses that are amenable to hand calculation (*eg.*, for estimating lock range [3]) are overly simplistic for most practical designs; for example, they do not take dynamics, which are very important in determining PLL responses, into account. Therefore, the rôle of simulation in PLL design assumes much greater importance than for the design of simpler systems like op-amps.

• system-level simulation with behavioral models (or *macromodels*): Behavioral simulation using phase-domain macromodels is extremely important in PLL design [4], [5] because of the great speedups it offers over transistor-level full simulation. Existing behavioral simulation of PLLs relies largely on using *linear* models for most components, especially for the VCO phase macromodel. The main issue with VCO behavioral models is unacceptable loss of accuracy and predictive power. Although it has generally been assumed that linear VCO macromodels[1] [4], [6] are adequate for behavioral simulation of PLLs, it has recently been demonstrated that using them can lead to very serious prediction errors [7], especially in the presence of nonlinear transient effects such as those involved in the capture, lock acquisition, and slipping processes in PLLs. The predictive power of linear VCO models is particularly poor for advanced PLL designs that use feed-forward or injection-aided mechanisms to enhance performance [8]–[10], as we investigate in detail in this paper. (Section III explains these mechanisms and design techniques in more detail.)

• transistor-level circuit simulation: In view of the significant accuracy problems in hand- and behavioral-level analysis of PLLs, designers rely heavily on transistor-level circuit simulation in existing PLL design methodologies. Such full simulation has the great advantage that it is able to predict non-ideal and nonlinear effects accurately. Unfortunately, as practitioners are well aware, full simulation of PLLs is extraordinarily time consuming. For example, a single jitter simulation for an industrial PLL can take days. The reason for the inefficiency of full SPICE-level simulation of PLLs stems from the fact that loop dynamics are typically orders of magnitude slower than the oscillation frequency of the VCO, resulting in a classic fast/slow timescale situation, where very small simulation time-steps need to be taken over a very long total simulation period. Because the only option for accurate PLL simulation in today's methodologies is so slow, designers are often forced to ignore large parts of the design space or to skip important verification steps simply due to time pressure. It is mainly for this reason that PLL design tends to be particularly error prone.

In this paper, we present a fast, accurate and extremely effective methodology for designing any kind of PLL. Our methodology involves extensive use of behavioral simulation using *nonlinear VCO phase-domain macromodels* that are automatically generated via algorithm from transistor-level VCO circuits. Our use of nonlinear phase macromodels is motivated by recent work [7], [11] which has established their suitability for predicting a variety of advanced or non-ideal effects, such as injection locking, capture and acquisition transients, jitter due to power supply variations, *etc.*. The most important benefit of the proposed methodology is that it dispenses with the need for time-consuming transistor-level simulations to a much greater extent than previously possible. The nonlinear behavioral

[1] *ie.*, the VCO inside the PLL is modeled as a linear integrator [6].

0-7803-9451-8/06/$20.00

simulations employed instead produce results virtually identical in accuracy, while being orders of magnitude faster.

We demonstrate this methodology by applying it to the complete design of an advanced new injection-aided PLL. Using our new methodology, we are able to design the new PLL to lock three times faster than similar conventional PLLs, without having to make tradeoffs that sacrifice other performance metrics such as noise/jitter.

We show in this paper how our new design methodology results in significantly improved design creativity and productivity. Using the proposed PLL design methodology (implemented in MATLAB) we are able to accurately simulate transient capture/locking within approximately one minute. In contrast, transistor-level simulation in the same simulation framework takes about fifty minutes. This speedup also has a great impact on the overall work flow of a designer, because the flow of ideas and design decisions is significantly improved by fast simulation turnaround times. With accurate simulations completing in a few seconds or minutes instead of in hours or days, it becomes possible and convenient to investigate many more different design scenarios or parameter sets. If time for thinking and design decisions (based on information from prior simulation runs) is included, we estimate that a typical designer can run approximately 5–6 PLL simulations per hour using our behavioral methodology, as opposed to an entire day for the same level of productivity.

Crucial to the effectiveness of our methodology is the fact that the fast behavioral simulations we employ *do not appreciably sacrifice accuracy* relative to full SPICE-level simulation. To validate the methodology, we compare against full transistor level simulations and always achieve excellent match, implying that far fewer full transistor-level simulations are needed when our methodology is employed. We also explore conventional behavioral methodologies [6] and demonstrate although they are equally fast, they completely fail to predict correct results.

The remainder of the paper is organized as follows. In Section II, we briefly describe conventional PLL design methodologies. In Section III we provide background on advanced PLL design concepts such as injection locking. In Section IV, we describe our new PLL design methodology and in Section V, we apply it to an injection-aided PLL design and describe its benefits.

II. Conventional PLL design methodologies and limitations

Conventional PLL design methodologies for behavioral simulation of PLLs are typically based on linearized analysis around a locked state. Figure 1 depicts the structure of a simple PLL and its linearized phase-domain model when in lock. The phase/frequency detector (PFD) is modeled as a multiplier with a gain K_d, low-pass filter (LPF) with a transfer function of $F(s)$ and a voltage-controlled oscillator (VCO) as a linear integrator. When locked, the negative feedback

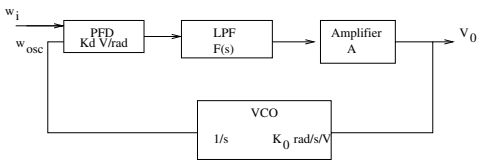


Fig. 1. Linear PLL Model

loop ensures that the frequency ω_{osc} from the VCO is identical to the input reference frequency ω_i. Using classical linear feedback control theory, the closed loop transfer function of the system can be derived to be

$$\frac{V_0}{\omega_i} = \frac{1}{K_0}\left(\frac{K_v F(s)}{s + K_v F(s)}\right), \tag{1}$$

where $K_v = K_d K_0 A$ is the loop gain, which determines the lock-in range. For a first-order loop filter

$$F(s) = \frac{1}{1 + \frac{s}{\omega_1}}, \tag{2}$$

the poles of the transfer function are

$$s = -\frac{\omega_1}{2}\left(1 \pm \sqrt{1 - \frac{4K_v}{\omega_1}}\right). \tag{3}$$

On comparing with a standard two-pole transfer function [12], the natural frequency ω_n and damping ratio ζ can be found to be

$$\omega_n = \sqrt{K_v \omega_1}, \qquad \zeta = \frac{1}{2}\sqrt{\frac{\omega_1}{K_v}}.$$

Similar results are also easily obtained for third and higher order loop filters [13]. Thus we see that the poles of the system function (which determine transient settling behavior for linear systems) have their real parts directly dependent on the loop filter bandwidth ω_1. ζ, which determines peaking in frequency response, is also dependent on ω_1. The loop filter bandwidth ω_1 also governs noise performance, while K_v determines the lock-in range and other performance metrics. Thus, it is apparent that even when only a linearized methodology is used, it can be a non-trivial optimization problem to achieve the right balance between different performance metrics.

But as demonstrated shortly, linearizations are at best a simplification valid in a narrow region around lock, so optimization of the linearized PLL is of limited value in any case. In particular, as has been demonstrated [7], PLL capture phenomena are inherently strongly nonlinear. In the methodology developed in this paper, we provide an effective means to take nonlinearities into account during PLL design.

III. Nonlinear effects in advanced PLL designs

Possibly the most important functional component inside any PLL is the voltage controlled oscillator (VCO). As is well known, amplitude-stable operation of oscillators is a fundamentally nonlinear process (*eg.* [14]). This leads to many important and fascinating effects in oscillators that are impossible to understand using linear concepts only.

A. Injection locking

One such effect, important for advanced PLL design as shown later, is *injection locking*. As the term implies, when an external weak signal is injected into an oscillator, then, under certain conditions, the oscillator's frequency changes to become identical to that of perturbing signal. Even if perfect lock to the external signal is not achieved, interesting and useful "frequency pulling" phenomena typically occur. Injection locking effects have been extensively studied (*eg.*, Adler [15], Kurokawa [16] and others [17]). However, several prior approaches to predict injection locking have relied on approximations like simplifying nonlinearities and neglecting higher order harmonics. Therefore these approaches are not able to predict results accurately when the circuit deviates from these assumptions significantly. In the methodology presented in this paper, we emphasize full consideration of nonlinearities to capture such phenomena accurately.

B. Injection-aided PLL design

There has been growing awareness in recent years that injection locking can be used to advantage in circuits that rely on phase synchronization. For example, injection locking has been used for quadrature generation in mixers [18]. In PLLs, injection locking has been applied to improve locking range, phase noise and jitter performance [8]–[10]. The increasing importance of new injection-aided PLL architectures has placed existing methodologies for PLL design [4], [5] under even greater stress, since injection locking is a

fundamentally nonlinear phenomenon that linearized approaches are completely incapable of predicting.

In this work, we also present new PLL design concepts enabled by our methodology. We design an injection-aided PLL prototype that significantly reduces capture and lock acquisition time. Further, we employ a soft switching approach to remove the injection locking path once the PLL has achieved lock, so as to enable complete freedom in optimizing other performance metrics (such as jitter). We believe that investigation and refinement of these ideas would have been impractical without our design methodology.

IV. Nonlinear phase macromodel based PLL design methodology

Our new methodology is based on nonlinear phase domain VCO macromodels. In our methodology, the VCO inside the PLL loop is modeled as a *nonlinear* element as compared to previous linear integrator models.

Existing linear VCO phase macromodels have the form

$$\dot{\alpha}(t) = K_{vco} b(t), \quad \text{or } \alpha(t) = K_{vco} \int b(\tau)\, d\tau, \tag{4}$$

where α is the phase deviation of the VCO caused by an external input (or perturbation) $b(t)$. In contrast, our nonlinear phase equation [14], [19] has the form

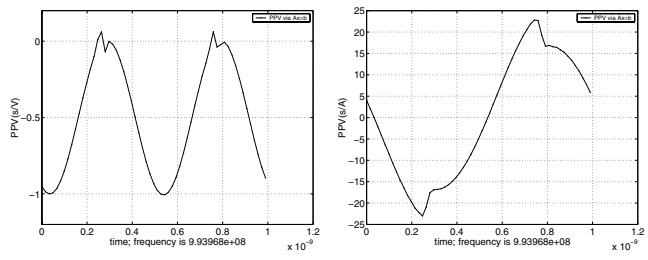
$$\dot{\alpha}(t) = v_1^T (t + \alpha(t)) \cdot b(t). \tag{5}$$

In this equation, $v_1(t)$ — called the perturbation projection vector (PPV) — is a vector of highly nonlinear, periodic, waveforms. Each node of the VCO has an associated PPV waveform component. These PPV components determine the effect of perturbations at the node on the output phase of VCO. It is this relationship that is captured by the nonlinear differential equation (5).

From a methodological viewpoint, the PPV waveforms for any oscillator can be easily extracted from a SPICE-level circuit of the oscillator, using numerical algorithms [19], [20]. It has already been established [7] that use of such macromodels leads to excellent prediction of capture/lock transients, cycle-slipping, static phase offsets and other effects in which injection locking/pulling plays a key rôle.

A. Design intuition from VCO PPV waveforms

A major benefit of our methodology is that examination of the PPV waveforms $v_1(t)$ can used to obtain direct design intuition and insight. Two PPV waveforms of an LC VCO are shown in Figure 2.



(a) (b)

Fig. 2. The PPV of control node and capacitor voltage node of the LC VCO

If a VCO is initially locked at ω_0 and an input frequency signal of frequency ω_1 disturbs the initially locked loop, then the output phase ϕ_{out} of the VCO can be expressed as

$$\phi_{out}(t) = \omega_0 (t + \alpha(t)). \tag{6}$$

Thus, if the VCO locks to ω_1, we have

$$\omega_1 t + \theta = \omega_0 (t + \alpha(t)), \tag{7}$$

or

$$\alpha(t) = \frac{\omega_1 - \omega_0}{\omega_0}\, t + \frac{\theta}{\omega_0}. \tag{8}$$

Therefore, for locking to occur, $\alpha(t)$ should change with time linearly, *ie.*, $\dot{\alpha}(t)$ *must have a constant DC value*.

From (5), we see that $\dot{\alpha}(t)$ is a multiplication of two waveforms: the PPV and the external input. If the PPV waveform is an AC waveform, then an AC input signal is required to obtain a DC component (thereby changing the VCO's frequency for injection lock); while if the PPV contains both AC and DC terms, then a DC signal can also make the VCO locked. The *frequency control node of VCOs is typically designed to have predominantly DC terms in its PPV component*. By injecting signals into other VCO nodes with strong AC terms in their PPVs, injection locking to aid lock acquisition can be usefully induced, as we describe further in Section V.

B. Nonlinear phase equation based design methodology

Before demonstrating our PLL design methodology by applying it to design a PLL in Section V, we first summarize its main steps:

1) Use existing phase domain behavioral models for the PLL's phase detector (PD) and frequency divider.
2) Model the loop filter, which is typically small, at the voltage level as a circuit or behavioral block.
3) Model the VCO using (5). Obtain the PPV $v_1(t)$ from the *full SPICE-level VCO circuit* via numerical algorithms [19], [20], thus setting up the nonlinear macromodel correctly. Crucially, identify all relevant inputs to the VCO, including the traditional control, auxiliary inputs like injection locking feeds, power supply and ground nodes (for jitter), *etc.*. Examine the PPV components of these nodes with a view to exploiting them during design.
4) Compose the PLL behavioral model using the above blocks and use it for simulations. Re-extract the VCO phase macromodel if changes are made to the internal circuitry of the VCO during design.
5) To regenerate voltage-domain waveforms from phase-domain ones, retain the VCO steady-state obtained during PPV extraction. Regenerate voltage-domain outputs using the steady state waveforms as described in [7].

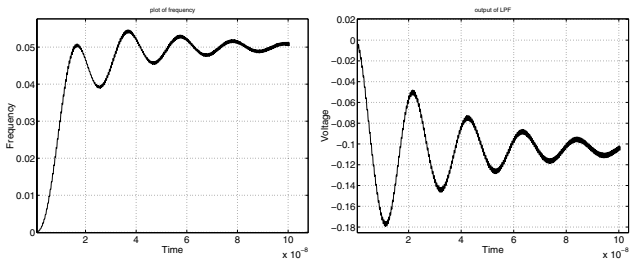
V. First-time-correct design of injection-aided PLL for fast lock acquisition

In this section, we use our methodology to design a novel injection-aided PLL with enhanced lock acquisition properties. We first apply behavioral simulation to the simple PLL shown in Figure 1. Then, applying the new design methodology and leveraging design intuition gained from each step, we improve the design in steps. After finalizing the design, we compare predictions from our behavioral simulations against full simulation to confirm correctness of our design.

A. Simulation of a Basic PLL

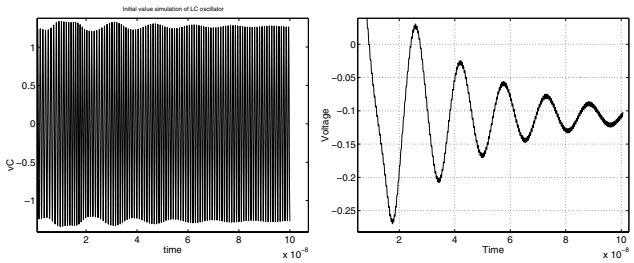
First, we simulate transient responses in the simple PLL loop. The (initially locked) VCO frequency f_0 is chosen to be 1Ghz and the loop filter bandwidth is taken to be 15 Mhz. We compare the step response of the PLL obtained using our methodology with that from a linear methodology, as well as against full SPICE-level simulation.

We inject a reference frequency signal of 1.05 f_0 and simulate the capture/lock transients of PLL until it reaches steady state. Figure 5 and Figure 3 depict lock acquisition transients, as simulated by linear



(a) (b)

Fig. 3. VCO frequency shift and control node voltage waveform using nonlinear macromodel



(a) (b)

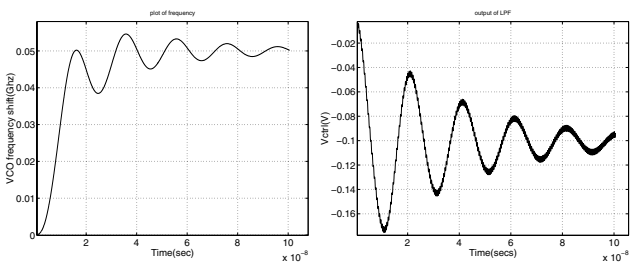
Fig. 4. VCO frequency shift and control node voltage waveform using full simulation

and nonlinear macromodels. As is apparent from the figures, the VCO frequency tracks the reference frequency after approximately 100 T (where T is the time period of the oscillator), when transients die out and the frequency shift settles to a constant factor of 0.05. Full simulation verifies that results from both macromodel simulations are approximately correct. The control node voltage waveforms also show good matches with that of full simulation – settling finally to -0.1 volt, close to estimates from hand calculations.

Thus, we see that for the simple PLL loop perturbed from lock, linear as well as nonlinear VCO macromodels work well. This is not surprising, since linearization is relatively valid for small perturbations from lock. The use of such macromodels provides approximately a 50 times speed up over full simulation.

B. Improving settling time response using injection locking

Using standard PLL design techniques, it is difficult to reduce the PLL's settling time without sacrificing aspects of noise performance. Improving loop settling time requires increasing loop filter bandwidth ω_1 (as described in Section II), thus resulting in more mixer noise propagating to the VCO input.



(a) (b)

Fig. 5. VCO frequency shift and control node voltage waveform using linear macromodel

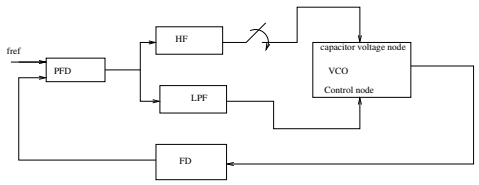
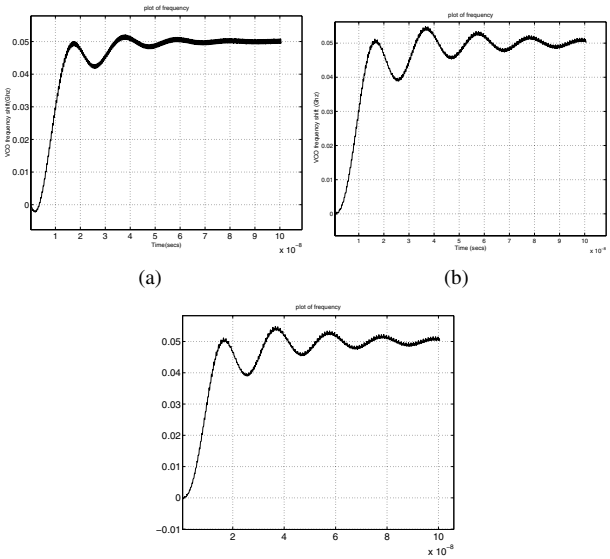


Fig. 6. New PLL Design



(a) (b)

(c)

Fig. 7. VCO frequency shift using different injection locking paths

To avoid having to make this undesirable tradeoff, we next investigate how an additional path that excites injection locking can be used to improve settling response, without changing the loop filter. This technique is motivated by observing strong AC terms in the PPV components of internal VCO nodes, *eg.*, the PPVs shown in Figure 2. The extra HF injection locking path is shown in Figure 6, immediately following the mixer. The rôle of the switch (which is normally closed during lock acquisition) will be explained shortly.

C. Injection to VCO capacitor node with different injection levels

In search of an effective injection locking path, we try three different injection locking signals to the capacitor voltage node of VCO: injection of the reference signal itself, injection of the full phase detector output signal, and a high-pass filtered version of the phase detector's output signal. We first attenuate the injection to make it one tenth of oscillator's free-running amplitude. We apply this injection to the capacitor voltage node of VCO. Results from behavioral simulations of this setup are shown in Figure 7.

From Figure 7, we see that for the case of injection of the reference frequency signal, the settling response of the PLL loop is improved, while in the other two cases injection has little effect on capture/lock transients of PLL loop. Keeping in mind that different injection levels lead to different levels of locking and pulling (as described in [11]), we try other injection signal levels. We increase injection signal levels to be comparable of that of oscillator signal and inject them into the capacitor voltage node of the VCO.

As we see from the simulation results in Figure 8, injection of the high-pass filtered phase detector output leads to considerable improvement in the settling time of the PLL loop. Injection of the reference frequency also speeds up the PLL's settling response. Although direct injection of the phase detector output also improves

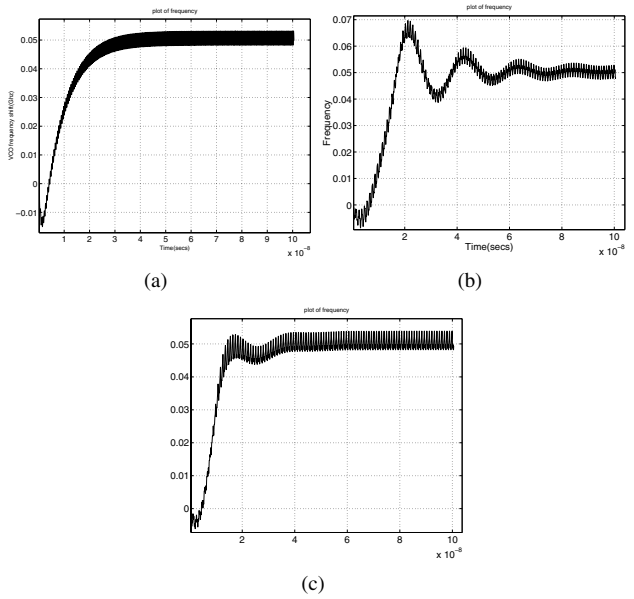


Fig. 8. VCO frequency shift using different injection locking paths

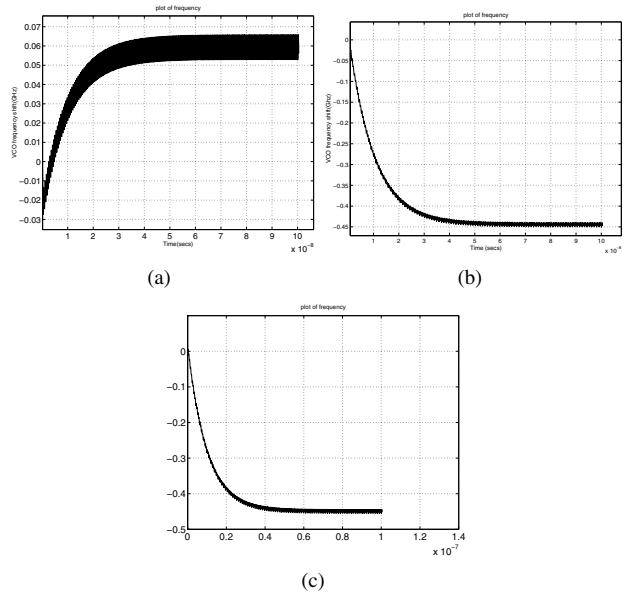


Fig. 9. VCO frequency shift using different injection paths

the PLL's settling response, the effect is less than by the other two paths.

Encouraged by these observations, we further increase the injection signal level, to five times that of the oscillation amplitude. However, we now observe from Figure 9 that this results in the oscillator's locking to a completely different frequency (a subharmonic of the reference frequency).

After experimenting with several other injection levels, we are able to find optimal injection paths and an optimal injection signal level, which leads to a speed up of three in settling-time response, compared to the PLL without injection-aided locking. These optimal paths are injection of reference frequency signal and injection of high pass filtered phase detector output with injection signal level comparable to oscillator signal.

It is worth mentioning here that conducting one such experiment with full SPICE level simulation takes approximately 50 minutes, as compared to about 1 minute for nonlinear behavioral simulation. As mentioned earlier, this speedup is crucial for enabling new design insights and ideas. We also emphasize that the above experiments, involving PLL design space exploration, could not have been carried out using a traditional linearized PLL design methodology [6]. Erroneous simulation results obtained by using linear macromodels (for the case of Figure 8) are shown in Figure 10 below. As already noted, linear models cannot account for effects such as injection locking.

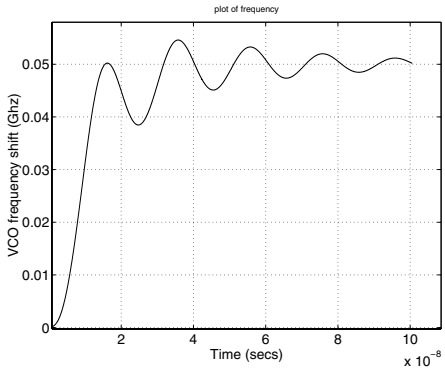


Fig. 10. The Plot of VCO phase shift using linear macromodel simulation for new PLL

D. Switching to disable injection aided operation after lock

Once the PLL has acquired lock, it can be desirable to remove the injection locking path from the loop for complete freedom in optimizing other performance metrics (such as jitter). This enables very easy augmentation of existing PLL designs to employ our injection-aided lock acquisition technique as described above.

To remove the injection, we cut the injection path using the switch shown in Figure 6. We first employ hard (abrupt) switching once the PLL has acquired lock. As we see from Figure 11, as soon as the switch is turned off, the PLL response shifts to that of the simple PLL with no injection path. This scheme fails as it increases the settling time. The reason stems from that the static phase offset for the injection-aided PLL is very different from that for the PLL without injection; abruptly removing the injection results in the PLL's losing lock again.

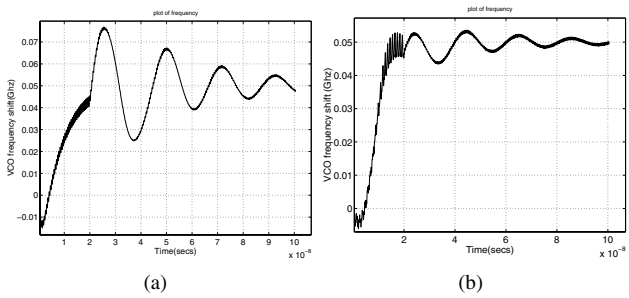
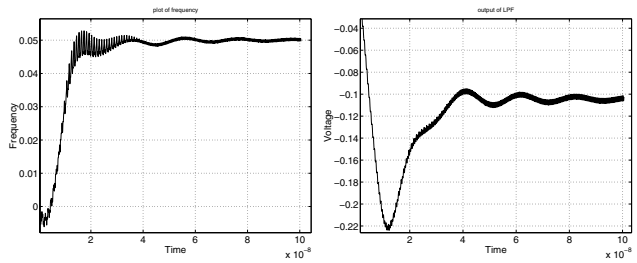


Fig. 11. Hard switching simulation for new PLL

In order to remove the injection while ensuring that the "regular" PLL loop always remains in lock, we next try a soft switching approach, *ie.*, taking the injection locking path out "slowly". As seen in Figure 12(a), the injection locking mechanism gradually relinquishes control of the VCO's frequency to the normal VCO control node and the static phase offset changes smoothly, without loss of lock at any point, to the value it has in the absence of injection locking.



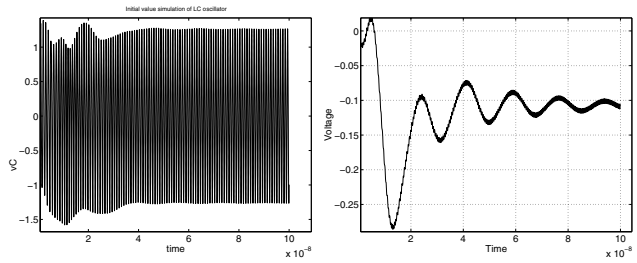
(a) (b)

Fig. 12. Soft switching simulation results: frequency offset and LPF output

Thus we see that injection of the high-pass filtered phase detector output improves PLL settling time to approximately one-third that possible via linear design techniques. Furthermore, using soft switching, this is achieved without affecting any other performance metrics.

E. Final verification against full SPICE-level simulation

To verify that the design indeed functions correctly, we compare the results of behavioral simulation of the final design against full SPICE-level simulation. The full SPICE level simulation voltage waveform,

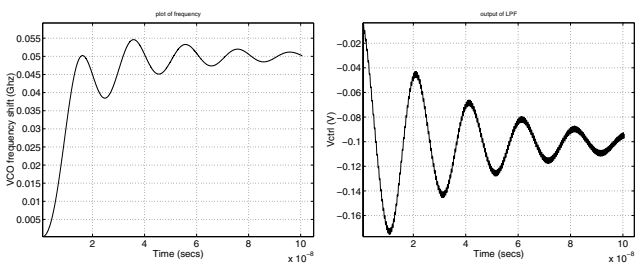


(a) (b)

Fig. 13. Full simulation results: VCO output voltage, LPF output

as shown in Figure 13(a), shows a constant envelope after a time period of about 40 cycles, as predicted by the macromodel-simulated VCO phase shift in Figure 12(a). Comparing this with the voltage waveform of the simple PLL loop in Figure 4(a) (which features a constant envelope in about 100 cycles), we confirm that the new PLL design settles approximately 3 times faster.

For further comparison, we also run behavioral simulations of the new PLL with linear VCO behavioral models used in existing PLL design methodologies. As shown in Figure 14, we find that they still represent the same transient behaviour as that of the simple PLL with no injection. These results confirm that linear macromodel based methodologies completely fail for such designs.



(a) (b)

Fig. 14. Simulations with linear macromodel for new PLL

VI. Conclusions

We have presented a new PLL design methodology based on nonlinear VCO macromodels. Using the new methodology, we have successfully designed a new type of PLL which exploits injection locking to speed lock acquisition by a factor of 3. The behavioral simulations used in our methodology run about two orders of magnitude than SPICE-level simulations, while retaining excellent quantitative and quantitive accuracy. This leads to design productivity improvements of an order of magnitude or greater. We have also shown how existing methodologies for PLL design, which do not account for VCO nonlinearities, would have been inadequate for this design. We anticipate that adoption of our methodology in industrial PLL design will significantly cut the time and cost of obtaining correctly functioning silicon.

References

[1] J.L. Stensby. Phase-locked loops: Theory and applications. CRC Press, New York, 1997.

[2] U.L.Rohde. Microwave and wireless synthesizers: Theory and design. Wiley,Hardcover, 1997.

[3] S.H.Lewis P.R.Gray, Paul J.Hurst and R.G.Meyer. Analysis and design of analog integrated circuits. John Wiley and Sons,Inc., 2001.

[4] K. Kundert. Predicting the Phase Noise and Jitter of PLL-Based Frequency Synthesizers. www.designers-guide.com, 2002.

[5] A. Demir, E. Liu, A.L. Sangiovanni-Vincentelli, and I. Vassiliou. Behavioral simulation techniques for phase/delay-locked systems. In Proceedings of the Custom Integrated Circuits Conference 1994, pages 453–456, May 1994.

[6] M. Takahashi, K. Ogawa, and K.S. Kundert. VCO jitter simulation and its comparison with measurement. In Proceedings of Design Automation Conference 1999, pages 85–88, June 1999.

[7] X.Lai,Y.Wan and J.Roychowdhury. Fast PLL simulation using nonlinear VCO macromodels for accurate prediction of jitter and cycle-slipping due to loop-nonidealities and supply noise. In Proc. Asia South-Pacific Design Automation Conference, January 2005.

[8] Peter K.Runge. Phase-locked loops with signal injection for increased pull-in range and reduced output phase jitter. Communications, IEEE Transactions on, 24(6), June 1976.

[9] Kwing F.Lee B.Razavi and Ran H. Yan. Design of high speed, low-power frequency dividers and phase-locked loops in deep submicron cmos. Solid-State Circuits, IEEE Journal of, 30(2), February 1995.

[10] T.Berceli S.Kudszus, M.Neumann and W.H.Haydl. fully integrated 94-ghz subharmonic injection-locked pll circuit. IEEE Trans. Microwave Theory Tech., 10(2), February 2000.

[11] X.Lai and J.Roychowdhury. Capturing injection locking via non-linear phase domain macromodels. IEEE Trans. Microwave Theory Tech., 52(9), September 2004.

[12] J.D.Powell G.F.Franklin and A.Naeini. Feedback control of dynamic systems. Pearson Education, 2002.

[13] I.V.Thompson and P.V.Brennan. Fourth-order pll loop filter design technique with invariant natural frequency and phase margin. IEE Proc. Circuits Devices Syst., 152(2), April 2005.

[14] X.Lai and J.Roychowdhury. Automated Oscillator macromodelling techniques for capturing amplitude variations and injection locking. In Proc.IEEE International Conference on Computer-Aided Design, November 2004.

[15] R. Adler. A study of locking phenomena in oscillators. Proceedings of the I.R.E. and Waves and Electrons, 34:351–357, June 1946.

[16] K. Kurokawa. Injection locking of microwave solid state oscillators. Proc. IEEE, 61:1386–1410, October 1973.

[17] M. Armand. On the output spectrum of unlocked driven oscillators. Proc. IEEE, 57:798–799, May 1969.

[18] P. Kinget, R. Melville, D. Long, and V. Gopinathan. An injection-locking scheme for precision quadrature generation. IEEE Journal of Solid-State Circuits, 37(7):845–851, July 2002.

[19] A. Demir, A. Mehrotra, and J. Roychowdhury. Phase noise in oscillators: a unifying theory and numerical methods for characterization. IEEE Trans. on Circuits and Systems-I:Fundamental Theory and Applications, 47(5):655–674, May 2000.

[20] A. Demir and J. Roychowdhury. A reliable and efficient procedure for oscillator ppv computation, with phase noise macromodelling applications. IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 22(2):188–197, February 2003.