Behavioral Simulation Techniques for Phase/Delay-Locked Systems

Alper Demir Edward Liu Alberto L. Sangiovanni-Vincentelli and Iasson Vassiliou

Department of Electrical Engineering & Computer Sciences
University of California, Berkeley, CA 94720

Abstract

This paper presents behavioral simulation techniques for phase/delay-locked systems. Numerical simulation algorithms are compared and the issue of numerical noise is discussed. Behavioral phase noise simulation for phase/delay-locked systems is described. The role of behavioral simulation for phase/delay-locked systems in our top-down constraint-driven design methodology, and in bottom-up verification of designs, is explained with examples. Accuracy and efficiency comparisons with other methods are made. Simulation techniques are described in the framework of phase/delay-locked systems, but simulation methodology and the results attained in this work are applicable to the behavioral simulation of mixed-mode nonlinear dynamic systems.

1. Introduction

Phase-locked loops (PLLs) and delay-locked loops (DLLs) are an important part of the class of mixed-mode nonlinear dynamic systems which includes PLLs, DLLs, oversampling data converters, switching power supplies. PLLs and DLLs have many applications; they are used in receivers, clock generators [3], clock recovery [4], data synchronization [5], etc.

In general, mixed-mode nonlinear dynamic circuits are stiff systems. Traditional simulators such as SPICE spend large amounts of CPU time on computing the accurate waveforms of the digital circuits, yet the waveforms of the much slower analog circuits determine the system behavior. Consequently, traditional simulators take too much time to compute the system behavior by simulating the whole system over the interval required by the analog circuits.

When a system is originally conceived in our top-down constraint-driven hierarchical design methodology [1], the realization of the digital circuits, as well as the details of the analog circuits, have not been decided yet. So, an accurate circuit simulation is neither possible nor desired. Therefore, designers desire to efficiently obtain the analog waveforms and the *timing* of the digital waveforms rather than the actual digital waveforms.

Key specifications for PLLs are acquisition time, capture range, lock range and phase noise/jitter. Phase noise is a major and very important PLL specification. For instance, this noise can cause data errors in communication systems, create inaccurate readings in instrumentation systems such as spectrum analyzers. Prediction of phase noise through simulations is crucial at the early stages of the top-down design. Phase noise simulation imposes tight restrictions on the numerical simulation algorithms to be used. *Numerical noise* created by the simulation algorithm should be negligible when compared with the phase noise to be computed for accurate results.

In this paper, in Section 2 below, we compare several algorithms for their numerical noise behavior and accuracy-efficiency trade-offs. In Section 3, behavioral phase noise simulation is discussed. In Section 4, use of behavioral simulation in bottom-up verification of designs is explained. The role of behavioral simulation for phase/delay-locked systems in our top-down constraint-driven design methodology is elucidated in Section 5.

2. Numerical Algorithms

Behavioral models of circuit components such as voltage-controlled oscillators (VCOs), phase detectors (PDs), and phase/frequency detectors (PFDs) which operate only on timing information have already been developed [2]. These are high-level mathematical representations of the components. They include differential equations, difference equations, transfer functions, statistical distributions, tables or arithmetic expressions [1]. Behavioral models which consist of these representations capture the input-output behavior of circuits, and they are independent of circuit architecture. They describe the first-order behavior of the circuit as well as the second-order analog effects in order to get a realistic idea of the performance of the overall system. With these models, a wide variety of phase/delay-locked systems can be simulated.

Simulation of systems, which are composed of the interconnection of such behavioral models, can be done by using a modified integration method. Digital circuits are simulated with an event-driven simulator, while the analog circuits are simulated with a differential equation solver. By interfacing the analog simulator with the digital, using a protocol, the two simulators are synchronized. At the interface, integration must pause at breakpoints which correspond to the exact *times* when the digital signals switch. Because switching times are generally not known ahead of time, they must be solved for during the integration. For instance, the time-derivative of instantaneous phase in a square wave VCO model is expressed as,

$$\dot{\phi}(t) = 2\pi f(v(t)) \tag{1}$$

where $v(t)$ is the VCO control voltage, and $f(v)$ is, in general, a nonlinear function relating the effective frequency of VCO to the control voltage [2]. The waveform at the output of the VCO is represented by zero crossings, which occur when $\phi(t)$ takes values which are integer multiples of π. At a time point during the integration of (1), if a zero crossing occurs in between the previous time point and current time point, the integration algorithm should track back and hit the point where the zero crossing occurs, as illustrated in Fig. 1. There are two schemes to solve for the timing of switching events:

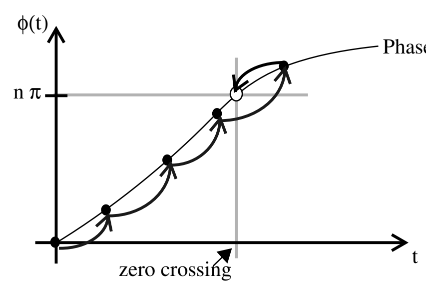

Fig. 1. Integration algorithm with zero crossings

- *Interpolation method.* Switching times are estimated based on a polynomial interpolation using the information from current and previous time points. For instance, with a first-order interpolation method, let a switching event be detected between the time points t and $t + \Delta t$, where Δt is the current time step size. Then, the time for the switching event is estimated by

$$t_s = t + \frac{\phi(t_s) - \phi(t)}{\phi(t + \Delta t) - \phi(t)} \qquad (2)$$

$$\phi(t_s) = n\pi \qquad n = 1, 2, 3\ldots$$

- *Iterative method.* Switching times are calculated using a bisection search (Fig. 2). If at time $t + \Delta t$, the phase $\phi(t)$ is larger

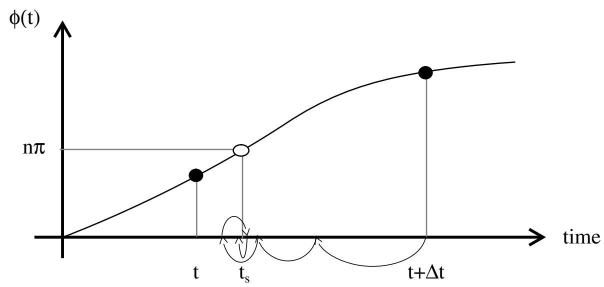

Fig. 2. Iterative search for switching times

than $n\pi$ while at t it was smaller, then there has been a switching event in between. To calculate the switching time, we divide the present time step by 2. Then we evaluate $\phi(t + \Delta t / 2)$. If this is still to the right of the switching time, we repeat the bisection procedure. If it is to the left, then we return the control to the integration algorithm. To prevent the algorithm from taking too much of a time step, we limit the maximum step size to $1/4$ of the last time step.

We have implemented an integration algorithm, including these schemes as options, to simulate phase/delay-locked systems. The integration algorithm is a variable time-step, variable order one based on the implicit backward differentiation formula [6][7]. First-order (linear) and second-order interpolation methods were implemented for estimation of the timing of switching events. The iterative method explained above, which can compute the timing of switching events to a desired accuracy, was also implemented. A behavioral model for a charge-pump PLL (Fig.3) [8], which consists of a ring-oscillator VCO, a PFD, a charge-pump and a second-order loop filter, was built to analyze the behavior of the numerical algorithms.

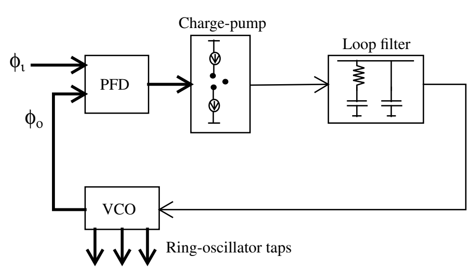

Fig. 3. Charge-pump PLL

The algorithms were compared by doing phase noise simulations of the above PLL. (Please refer to Section 3 for a discussion of behavioral phase noise simulation). Accuracy, in this context, is used to mean the accuracy of the solutions for the timing of switching events.

First, a comparison of the first-order and second-order interpolation methods was made, which was in favor of the first-order method with a very slight accuracy difference. Then, the first-order interpolation method was compared with the iterative method. There are basic differences in the behavior of these methods. The accuracy of the iterative method can be set to any desired value. However, there is *no* control on the accuracy of the interpolation method. During an unlocked state, while PLL is acquiring the input, the accuracy of the interpolation method decreases considerably. When PLL is "in lock", accuracy is inversely proportional to the noise level in the circuit. This is, in fact, a desired property, because more accuracy is needed for smaller noise levels. On the other hand, the accuracy of the iterative method stays at the desired/set level regardless of the state or noise level of the circuit. Fig. 4 shows the response of the above PLL to a frequency step at its input in the presence of noise. Even after acquisition, deviations around the mean value occur because of the noise in the circuit. Fig. 4 also shows the error in the solution of the timing of switching events for the interpolation method and the iterative method. The error for the iterative method in Fig. 4 is not visible on the plot that uses the same scale as the plot for the interpolation method, because the error for the iterative method was several orders of magnitude smaller. The details can be seen in the blow-up plot.

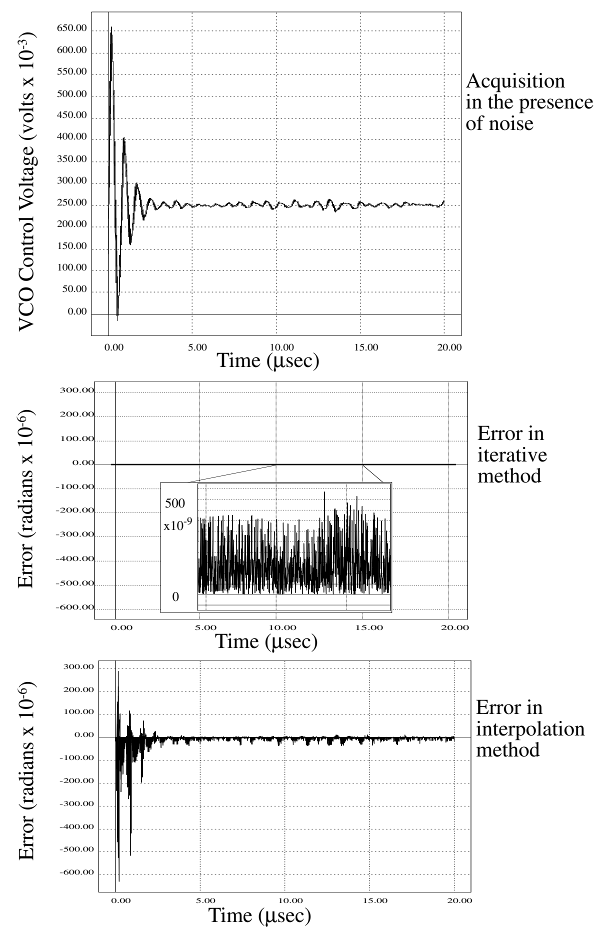

Fig. 4. Acquisition in the presence of noise

For the PLL considered above, the accuracy of the interpolation method was found to be sufficient for doing phase noise simulations when the circuit is "in lock". The results obtained by the interpolation method were verified by using the iterative one.

Although this was the case for the PLL considered, there is no guarantee on the accuracy of the interpolation method in general. On the other hand, the accuracy of the iterative method does not depend on the type, state or noise level of the circuit. The accuracy can also be set to a desired level where numerical noise will not interfere with the results.

3. Phase Noise Simulation

Phase noise/jitter is a very important specification for phase/delay-locked systems. Design for low phase noise is probably the most challenging part of the design of such systems. The difficulty of design for low phase noise is amplified by the lack of efficient simulation tools for its prediction. Prediction of phase noise through simulations is crucial at the early stages of the top-down design. Any modeling or design methodology, and simulation tools, which do not address this problem are not suitable for high performance phase/delay-locked system design.

The models we use in behavioral simulation include statistical second-order effects [2]. Analysis of a phase/delay-locked system in the presence of noise, when the system is *not* "in lock", is a hard problem. Fortunately, almost all of the time, a detailed observation of the system, when it is in the unlocked state, is not required as long as the loop is *guaranteed* to lock on to the input in the specified amount of time. This can be assured by simulations. At this point, it is assumed that the system is "in lock" for phase noise simulation. The expression, "in lock", is used in a time-averaged sense here, not in the instantaneous sense. For instance, for a PLL (say with a ring-oscillator VCO, charge-pump and PFD), instantaneous frequency of the VCO will have variations around a mean value because of the noise in the circuit. The mean value of the VCO frequency will be equal to the input frequency. It is assumed that phase noise performance of the circuit is above a level where it does not lose lock (in a time-averaged sense). This is a reasonable assumption, because if it does not hold, that means that we are confronted with a bad design which has deterministic performance problems. It is important to understand that "in lock" assumption does not mean we are using a linear model for the circuit. Linearity assumption might be validated for very low noise levels, but higher noise levels will excite the nonlinearities. For behavioral phase noise simulation, we use the full nonlinear models of the components, which are described in [2].

The method that is used for phase noise simulations is Monte Carlo in time domain. The CPU time disadvantage of the Monte Carlo method disappears because of the high efficiency of behavioral simulation. As discussed in Section 2, having a numerical algorithm with *controllable* numerical noise makes it possible to predict phase noise accurately. In behavioral phase noise simulation, circuit, which is "in lock", is simulated over a period of time while the delay cells add jitter. Then, the characteristics of jitter at the output is calculated by a statistical analysis over that time period. An example about phase noise simulation will be given in Section 5.

4. Behavioral Simulation in Bottom-Up Verification

The traditional approach to the bottom-up verification of phase/delay-locked system designs is to use a transistor-level simulator such as SPICE. Transistor-level simulation of a phase/delay-locked system takes too much time to be practical, because the system is stiff. Macromodeling these systems in a circuit simulator was proposed to circumvent the efficiency problems of transistor-level simulation [9][10]. This approach also results in impractical simulation times, because many circuit elements are needed for accurate models. Still, the accuracy attained by the macromodeling approach is not as high as needed for verification of designs. On the other hand, behavioral simulation achieves the goal of verifying complex system behavior efficiently. This is made possible by accurate statistical behavioral models which include analog second order effects. Evaluations of behavioral models are fast which results in an efficient system simulation.

Bottom-up verification using behavioral simulation is done in two steps as illustrated in Fig. 5:

- *Set up the behavioral models for the components.* The model parameters are extracted using SPICE from the transistor level description of components.
- *Simulate the system in time-domain using the behavioral simulator to calculate the performance measures*. Behavioral simulation is much faster than circuit simulation.

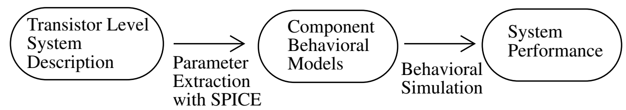

Fig. 5. Two-step verification with behavioral simulation

To illustrate this procedure, component behavioral models for a bipolar PLL [11] were set up, and behavioral simulation was used to analyze the acquisition characteristics. Fig. 6 shows SPICE domain extraction of the relation between the effective frequency of VCO and the control voltage ($f(v)$ in (1)). Other model parameters are

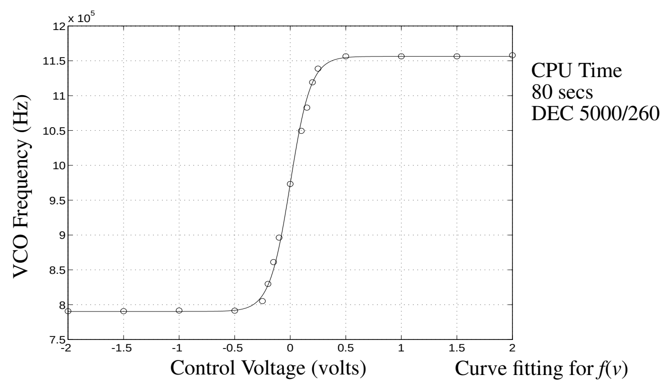

Fig. 6. Extraction with SPICE

extracted in a similar way from the transistor level description. Model extraction is done only once for a circuit. Then created models are used in many behavioral simulations to analyze the acquisition characteristics, stability, timing jitter and others. Fig. 7 shows the response of the modeled PLL to a frequency step at its input (both SPICE and behavioral simulation).

5. Behavioral Simulation in Top-Down Design

Behavioral simulation and modeling reduces the design time considerably in a top-down approach. Designers can verify system design early, with behavioral models of system components, before investing time in detailed circuit implementation. Design space exploration can be done efficiently. In our top-down, constraint-driven design methodology [1], given a set of circuit specifications (circuit characteristics, the design rules, the technology, and user options), a *mapping* is made to schematics or layout. Behavioral simulation and optimization are used for architecture selection. Once a particular architecture is chosen, behavioral simulation and optimization are used to map the architecture and circuit specifications to the detailed specifications of components or subblocks.

The role of behavioral simulation in top-down design of a

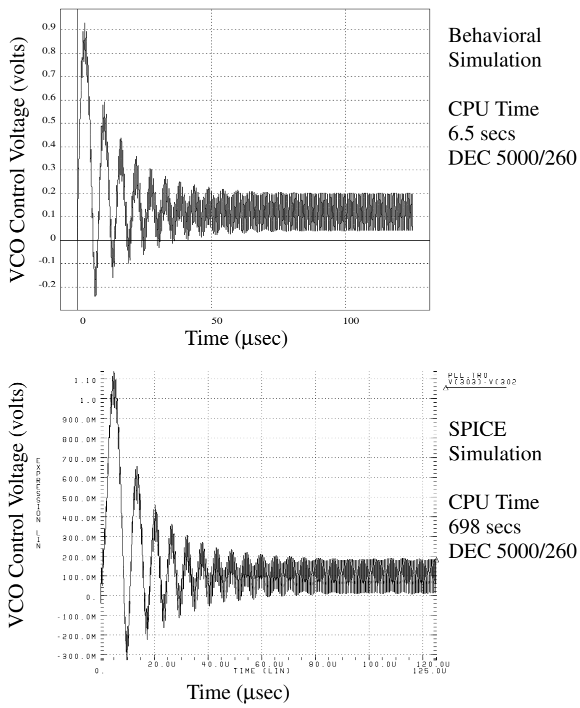

Fig. 7. Behavioral and SPICE simulation of acquisition

phase/delay-locked system will be illustrated with a multi-phase clock generator. A PLL with a ring-oscillator VCO or a DLL can be used to generate multi-phase clocks (Fig. 8). These two architectures were compared for their phase noise performance using behavioral simulation. Fig. 9 shows the relationship between clock jitter and percentage delay cell jitter for a given design of both architectures. Ring-oscillator VCO for PLL, as well as the delay line of DLL, has 5 delay cells. Reference clock frequency is 50 Mhz.

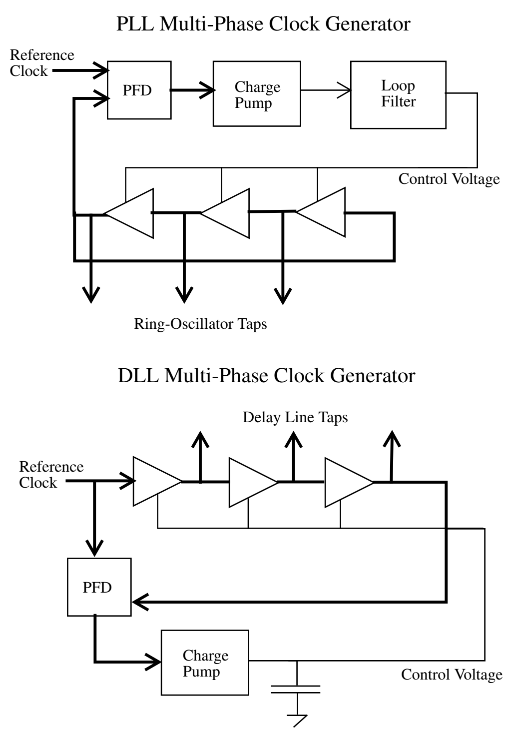

Fig. 8. PLL and DLL multi-phase clock generator

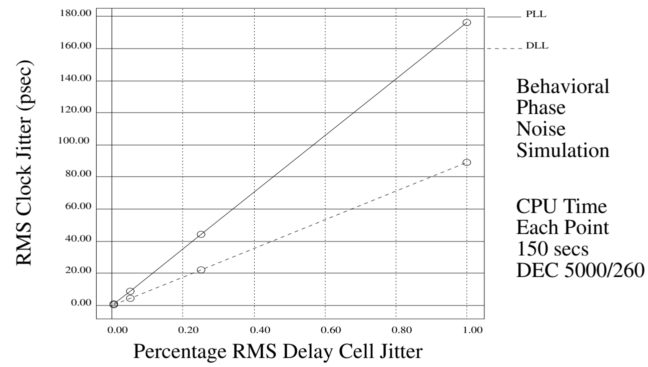

Fig. 9. Clock jitter for PLL and DLL clock generator

From Fig. 9, we conclude that DLL has better phase noise performance, when compared with a PLL, for fixed percentage delay cell jitter. Then, the relationship between clock jitter and percentage delay cell jitter is used to predict amount of jitter allowable an a delay cell, given a clock jitter allowance. In this way, the clock jitter constraint is mapped on to the delay cell jitter constraint.

6. Conclusions and Future Work

We have presented behavioral simulation techniques for phase/delay-locked systems. Numerical algorithms and behavioral phase noise simulation have been described. The role of behavioral simulation for phase/delay-locked systems in our top-down constraint-driven design methodology, and in bottom-up verification of designs, has been explained with examples. We are in the process of top-down, constraint-driven design of a clock generator circuit for a RAMDAC using the simulation tools we have presented. As we move down the hierarchy of design, we will create new tools and/or modify the existing ones as need arises.

Acknowledgements

This project is supported by DARPA and MICRO. Their support is gratefully acknowledged.

References

[1] H. Chang, A. Sangiovanni-Vincentelli, F. Balarin, E. Charbon, U. Choudry, G. Jusuf, E. Liu, E. Malavasi, R. Neff and P. Gray, "A top-down constraint-driven design methodology for analog integrated circuits", *IEEE CICC,* May 1992.

[2] E. Liu and A. Sangiovanni-Vincentelli, "Behavioral representations for VCO and detectors in phase-lock systems", *IEEE CICC,* May 1992.

[3] I.A. Young, J.K. Greason and K.L. Wong, "A PLL clock generator with 5 to 110 MHz of lock range for microprocessors", *IEEE Journal of Solid-State Circuits,* vol. 27, No. 11, November 1992.

[4] T.H. Lee and J.F. Bulzacchelli, "A 155-MHz clock recovery delay- and phase-locked loop", *IEEE Journal of Solid-State Circuits,* vol. 27, No. 12, December 1992.

[5] K.M. Ware, H-S. Lee and C.G. Sodini, "A 200-MHz CMOS phase-locked loop with dual phase detectors", *IEEE Journal of Solid-State Circuits,* vol. 24, No. 6, December 1989.

[6] T.L. Quarles. *Analysis of Performance and Convergence Issues for Circuit Simulation*. Ph.D. Thesis. (Call No: T7.6 1989 Q82 ENGI) U.C. Berkeley, April 1989.

[7] EECS 219. *Circuit Theory and Computer-Aided Analysis.* Lecture Notes. U.C. Berkeley. Fall 1991.

[8] F.M. Gardner, "Charge-pump phase-lock loops,", *IEEE Transactions on Communications,* vol. COM-28, No. 11, November 1980.

[9] M. Sitkowski, "The macro modeling of phase-locked loops for the SPICE simulator", *IEEE Circuits and Devices Magazine*, vol. 7, No. 2, March 1991.

[10] E. Tan. *Phase-Locked Loop Macromodels*. Masters' Thesis. (Call No: T7.49.1990 T283 ENGI) U.C. Berkeley, August 1990.

[11] P.R. Gray and R.G. Meyer. *Analysis and Design of Analog Integrated Circuits* p.624. Second Edition. John Wiley & Sons. 1984.