Abstract

Cycle Domain Simulator for Phase-Locked Loops

Norman James

October 1999

As computers become faster and more complex, clock synthesis becomes critical. Due to the relatively slower bus clocks compared to the processor, it is necessary to use phase-locked loops (PLL's) for frequency multiplication and phase alignment of the clocks.

A computer design environment focuses mainly on digital design. A PLL being composed of both digital and analog components does not match this type of design environment. Available design tools such as SPICE are adept at PLL simulations; however, they require long simulation time and are not well suited to the ways PLL's are used in computer systems. This literature survey explores current ways of simulating PLL's, and gives the foundation for developing a simulator that is specifically designed for PLL's.

INTRODUCTION

Traditionally, phase-locked loops (PLL's) implemented on a chip (microprocessor, DSP, ASIC, clock chip, etc.) are modeled with general-purpose simulators such as SPICE. PLL's are used in synchronous computer systems throughout the clock distribution to frequency multiply and phase align the primary clocks. They can be arranged in series or parallel and can be on a separate chip or integrated into another chip. The computer as a synchronous system implies that each replicated (synthesized) clock has a known phase and frequency relationship to its reference. This makes the phase relationship (phase error) an important parameter in chip-to-chip communication. Also since chips have an upper frequency limit to which they operate, knowing the frequency distribution is imperative. These phase and frequency errors are known as jitter. Jitter is the main parameter that any simulator should be able to predict.

There are several different simulators used in industry for PLL simulations, each one having its advantages and disadvantages. These include SPICE, EESof, Matlab, and analytical/behavioral models. The main aspects that make a PLL difficult to simulate are as follows:

- Two sources of feedback – main feedback ("loop" in phase-locked loop) and oscillator feedback [3]
- High frequency clocks in conjunction with low frequency time constants [1][2]
- Requires picosecond accuracy
- Digital components mixed with analog components (mixed-mode)
  - Digital – phase detector, divider
  - Analog – voltage-controlled oscillator (VCO), charge pump, loop filter

The main aspects that may a simulator useful for modeling PLL's are as follows:

- Simulation speed
- Modeling accuracy of CMOS processes
- Accurate jitter prediction (clock stability)
- Accurate lock-range prediction (frequency range over which the PLL will lock)

SPICE

SPICE is a well-known general-purpose time-domain circuit simulator [1][2]. The concern when simulating PLL's, is that there is a high frequency component such as the clock that must be simulated in conjunction with low frequency components, such as the lock time [1]. The clock period can be on the order of 1 ns and the time constants associated with the lock time (time required for PLL to phase/frequency lock on its reference) can be in the milliseconds.

To sample the 1 GHz square wave requires a sample roughly every 50ps for 1ms, it would require 20 million time points [2]. This is just for one node of the schematic. This in conjunction with doing a matrix solution for each time step for thousands of transistors is not feasible in a reasonable amount of time.

Although SPICE, given accurate models, can perform an accurate simulation, it can be very time consuming. The utilization of a voltage-controlled tuned circuit as the VCO that does not require feedback to produce an oscillating output is a method that can be used to reduce the simulation time [3], but still the number of data points and calculations required is immense.

The advantage of SPICE is that the models that accurately define a CMOS process are readily available. SPICE will typically always be used for determining the VCO frequency range and doing independent component tuning because of the superior model accuracy.

MATLAB

Matlab is an analysis tool published by MathWorks. It has toolkits available for analysis in areas such as DSP, communications, and control systems. Simulink is the graphical interface for building simulation models.

Included with the communications toolkit are charge-pumped PLL models. These models run fairly fast but still generate large amounts of data. Another drawback of Matlab is the included models would have to be significantly modified to simulate a PLL built in a real CMOS process. Matlab still has the notion of time-steps which leads to some of the same problems with SPICE, but doesn't have to deal with transistor level calculations. Clearly accuracy would suffer if the analytical model didn't mimic the behavior in silicon.

Efforts have been made to increase the accuracy of Matlab simulations by better modeling of the PLL blocks to reflect the CMOS process [4][5]. These efforts incorporate a CAD tool to create circuit prototypes of the functional block. The prototypes mimic the behavior of the block and the input and output load characteristics.

EESOF

EESof is an extensive software tool published by HP. It has many different modules to it including microwave, RF, high speed interconnect, and device modeling and extraction.

For PLL simulations, EESof uses a technique called circuit envelope simulation [2]. It accepts the input stimulus as RF carriers with time-varying complex envelopes (i.e. amplitude and phase modulations). The output solution is represented as a sum of the RF carriers and their harmonics, each with a time-varying complex envelope. Circuit Envelope has a fundamental advantage over time-domain simulators in that the time step-size need only be small enough to capture the bandwidth of the modulation envelope (which is about 30kHz), instead of the RF carrier (which is 1 GHz) [1].

The Circuit Envelope technique works well with the continuous time aspects of the PLL, but for components such as the digital phase detector would be more awkward.

PLL's in communication applications use sinusoidal phase detectors which is more easily used with EESof.

ANALYTICAL/BEHAVIORAL MODELS

Traditionally, the textbook s-domain PLL model has been used to model the behavior of the loop in the locked state [9]. The input and output waveforms are assumed to be sinusoidal and the phase detector is modeled as a linear analog multiplier with an inherent ideal low-pass filter [6]. Many charge-pumped CMOS PLL's do not behave exactly like the model represents.

Z-domain models for discrete-time PLL's have been developed as well. The z-domain model takes into account the sampled nature of the digital phase detector and accurately predicts the overall loop performance[6]. Although useful for predicting the input to output jitter relationship of the loop, the model is not a simulator. It would be extremely difficult to map it into the time-domain and use for spread spectrum and noise effects.

Other behavioral models have been developed using Analog Hardware Definition Language (AHDL). SprectreRF is a software design tool published by Cadence that uses AHDL to describe analog circuits in terms of their behavior [7][8]. Equations, state machines, or discrete circuit elements can be used to describe the PLL components in AHDL. The simulations are done in the time-domain similar to SPICE, so there is still the data explosion problem. Similar to AHDL, several custom simulators have been developed using C and other programming languages [10][11].

The main problem associated with all of the behavioral models mentioned is the fact that they work in a domain (time-domain, frequency-domain, s-domain, z-domain) that is not optimum for PLL's. Either the domain is not robust enough or not efficient enough.

PROJECT DIRECTION

Ideally, the simulation should relate back to the time-domain, since that is the world in which we live. The goal of this project is to develop a new domain that is well suited to PLL's (and clocks in general) and a simulator that uses this domain to perform robust and efficient simulations.

REFERENCES

[1] E. L. Acuna, J.P. Dervenis, A.J. Pagones, F. L. Yang, and R.A. Saleh, "Simulation Techniques for Mixed Analog/Digital Circuits," IEEE J. of Solid-State Circuits, vol. 25, no. 2, Apr. 1990, pp. 353-362.

[2] H. S. Yap, "Designing to Digital Wireless Specifications Using Circuit Envelope Simulation," HP EESof Application Note, 1995.

[3] W. E. Thain and J. A. Connelly, "Simulation and Modeling – an Improved VCO Macromodel," IEEE Circuits and Devices Magazine, vol. 84, July 1992, pp. 8-16.

[4] K. W. Current, J. F. Parker, and W. J. Hardaker, "On Behavioral Modeling of Analog and Mixed-Signal Circuits", Signals, Systems, and Computers, vol. 1, 1994, pp. 264-268.

[5] A. Albiol, N. Cardona, and J. Marti, "An Interactive Program for PLL Simulations," Frontiers in Education, 1992, p. 835.

[6] J. P. Hein and J. W. Scott, "z-Domain Model for Discrete-Time PLL's," IEEE Transactions on Circuits and Systems, vol. 35, no. 3, Nov. 1988, pp. 1393-1397.

[7] A. Phanse, R. Shirani, R. Rasmussen, R. Mendal, J. S. Yuan, "Behavioral modeling of a phase locked loop," Southcon/96, Conference Record, 1996, pp. 400-404.

[8] M. Takahashi, K. Ogawa, and K. S. Kundert, "VCO Jitter Simulation and it Comparison with Measurement," Design Automation Conference, 1999, vol. 1, pp. 85-88.

[9] D. Armaroli, V. Liberali, and C. Vacchi, "Behavioral Analysis of Charge-Pump PLL's," Circuits and Systems 1995, vol 2., 1996, pp. 893-896.

[10] B. A. A. Antao, F. M. El-Turky, R. H. Leonowich, "Mixed-Mode Simulation of Phase-Locked Loops," Custom Integrated Circuits Conference, 1993, pp. 8.4.1-8.4.4.

[11] P. Larssom, "A Simulator Core for Charge-Pump PLLs," Circuits and Systems II: Analog and Digital Signal Processing, vol. 45, no. 9, Sept 1998, pp. 1323-1326.