# Abstract

## Cycle Domain Simulator for Phase-Locked Loops

Norman James

December 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) for multiplication and phase aligning of the clocks.

A PLL is composed of both digital and analog components and is not modeled well in a design environment for digital systems. There are design tools available that are more adept for doing PLL simulations; however, they can be very costly and are still not suitable for the way PLL's are used in computer systems. While the literature survey discussed current methods for simulation, the goal of this project is to introduce a new simulator that is specifically designed for simulating PLL's used in computer systems.

## INTRODUCTION

Phase-locked loops (PLL's) are used in synchronous computer systems throughout the clock distribution to 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. Traditionally, PLL's implemented on a chip (microprocessor, digital signal processor, ASIC, clock chip, etc.) are modeled with general purpose simulators such as SPICE. Other simulators include 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
- High frequency clocks in conjunction with low frequency time constants
- Picosecond accuracy
- Digital components mixed with analog components (mixed-mode)
- Digital – phase detector, frequency divider
- Analog – voltage-controlled oscillator (VCO), charge pump, loop filter

A PLL simulator should:

- Run fast
- Model CMOS processes
- Predict jitter (clock stability)
- Predict lock range (frequency range over which the PLL will lock)

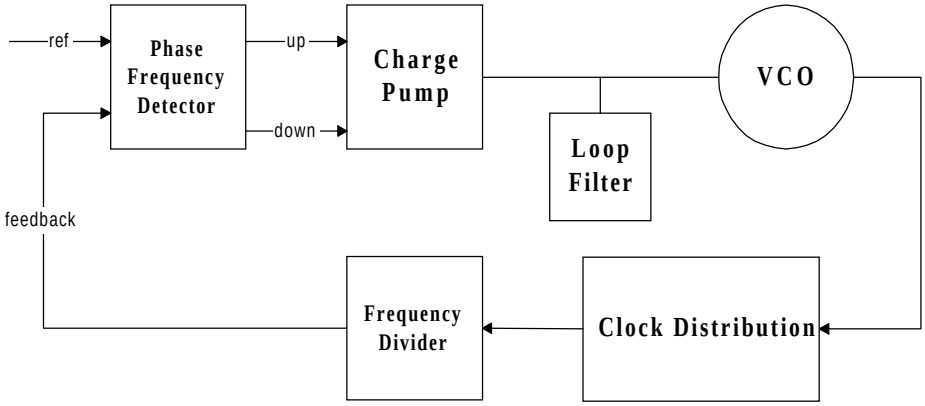


Figure 1: Phase-Locked Loop Block Diagram

This project introduces a simulator, known as CycleSim, that meets the outlined specifications and gives the designer a new tool for PLL applications. The report will first present a brief overview of PLL operation. Then CycleSim's operation is discussed and finally simulation results are presented.

## PLL OVERVIEW

In general, a PLL used in computer systems is called a charge-pumped PLL and is made up of the following components: phase-frequency detector (PFD), charge pump, loop filter, voltage-controlled oscillator (VCO), and frequency divider. The basic block diagram is shown in Figure 1.

Essentially, a reference clock is driven into one input of the PFD and a feedback clock is fed into the other input. The PFD outputs an up or down pulse depending on whether the feedback is leading or lagging the reference. The up and down pulses are proportional to the difference in phase of the two clocks. The pulses are translated into current by the charge pump, which either forces current into or out of the loop filter. A basic loop filter integrates the current and generates a voltage which is the control for the

VCO. If the PFD is generating up pulses, the control voltage is "pumped" up causing the frequency of the VCO to increase. The frequency divider, divides the VCO output by the desired bus to chip multiplication factor.

## CycleSim Overview

CycleSim is written in ANSI C++ and has been compiled under Windows NT and AIX. The simulator is neither a time-domain or frequency-domain simulator. It is a cycle-domain simulator. The basic premise is that if a PLL were considered a black box with one input and one output, both input and output would be clocks and the only significant aspect of these clocks is the period and phase. The amplitude aspects are of no importance since we are primarily concerned with the phase and frequency accuracy of the clock.

The simulator stores and works on rising edges only. This is equivalent to taking a sampled clock (from a simulator or hardware) and extracting the zero-crossing points which are essentially the points in time at which the waveform rises through some threshold (typically the midpoint). These points in time are referred to as time tags for the purposes of this report. To illustrate the nature of the time tags, Figure 2 is a plot of Simulation Iterations vs. Time Tags for clocks where one clock is three times faster than the other clock.

CycleSim has no notion of a time step size. For each iteration, the time tags for the multiple nodes are adjusted. For example, a simple simulation would contain three time tagged nodes, the reference clock, the VCO output, and a divider output. Since the reference frequency is fixed, the time tags simply increase each iteration by the period. The VCO is more complicated in that its period is constantly being adjusted by power

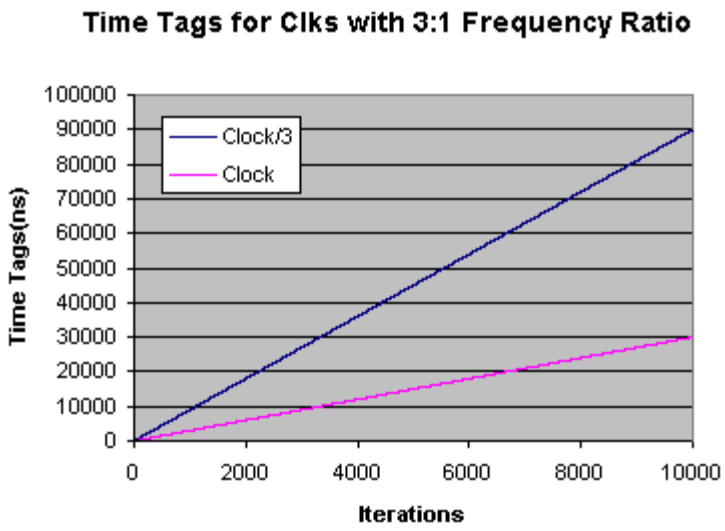


Figure 2: Time Tags illustration showing two clocks with a 3:1 frequency ratio

supply noise, the control voltage, and a feedfoward current port. The divider is simple because it just changes the period of the incoming time tags by the desired divider setting. The only circuit issues the simulator cares about are ones that adjust the time tags and that the time tags can easily be mapped into the time domain.

## SIMULATION RESULTS

### Simulation Speed

The simulations are several orders of magnitude faster than time-domain simulators such as SPICE. For 25,000 cycles, the simulator takes < 5 seconds to complete on a Pentium 133. Even scaled to slowest processors, the performance is still reasonable. Also, the number of data points created is the number of cycles times the number of nodes. Table 1 is a comparison of SPICE and CycleSim for a PLL simulation.

| | SPICE | CycleSim |
|---|---|---|
| **Simulation time (hours)** | 120 | 4[1] |
| **Number of data points (output node)** | 10,000,000 | 25,000 |
| **Accuracy (ps)** | 10 | .03[2] |

[1] Includes the time to perform SPICE simulations on individual components
[2] Accuracy of SPICE simulation on individual components

Table 1: Simulation comparison for a Phase-locked loop lock test

**Basics**

The first test was to make sure the PLL would lock phase and frequency. Figure 3 shows the first 400 cycles of a PLL simulation on a plot of period vs. cycle. This is a typical dampened sine wave response. The parameters could be adjusted to givevarious natural frequencies and dampening factors. The "jaggedness" of the line in the *y* direction is jitter once the loop is locked.

**Existing Model Correlation**

Another way to validate the simulator's behavior is to make sure it is obeying the *s*-domain equations relatively well. It will never match exactly due to the "digital" nature of the PFD [1]. The frequency of the oscillation can be measured to calculate the natural frequency. Using the basic PLL loop equations, the simulation results agree with the calculated parameters.

**Hardware Correlation**

A simulator is not much value unless it predicts the behavior of working hardware. Figure 4a is a period vs. cycle graph (also known as a cyclegraph) taken using

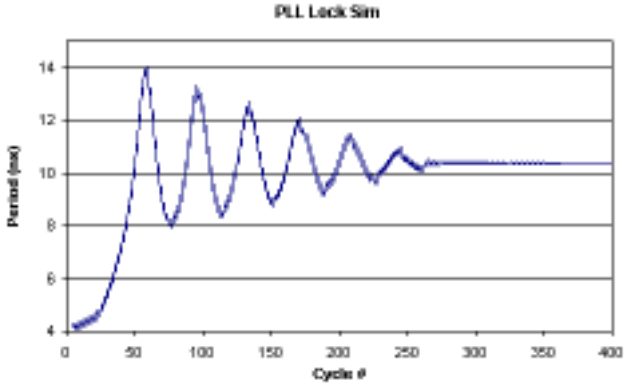


Figure 3: Lock simulation result

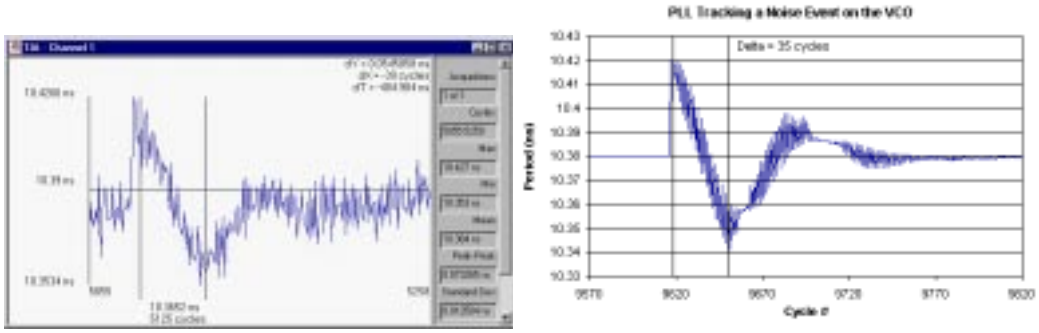


Figure 4: Noise response: (a) hardware (b) simulation

jitter measurement hardware. The graph is showing the response of the PLL output during a step change on the power of the VCO. Figure 4b shows the results of the simulator for the same event. The idea is to match the frequency of the oscillation and the amplitude.

## CONCLUSION

No simulator is perfect, but hopefully CycleSim makes the best compromises for PLL's. Using this simulator in conjunction with SPICE could be a very powerful tool for PLL designers. The fast simulation speed allows for new PLL topologies to be explored, along with taking the existing circuits to their limits. As the computers get faster, the accuracy of the clock becomes much more critical. The industry is starting to focus more

on jitter and noise issues that affect PLL's because of the smaller margins allowed in the system. Hopefully CycleSim can be used to design a sub-picosecond PLL that is impervious to noise.

## REFERENCES

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