MODELING PHASE-LOCKED LOOPS USING VERILOG

Jeffrey Meyer
Director of Engineering
Symmetricom, Inc.
3750 West Wind Blvd.
Santa Rosa CA 95403, USA

Abstract

An essential component of any mixed signal embedded system is a Phase-Locked Loop (commonly know as PLL). Almost every mixed signal system has one or more PLL in its block diagram. Phase-locked loops are used for a variety of tasks, like multiplying clock frequencies, generating precise clock phases, and generating complex RF modulated signals like phase modulation. Many modern field programmable gate array devices come with integrated PLL to multiply clocks or adjust the phase of clock outputs. Modeling PLLs has always been difficult because they are part analog and part digital. Circuits that are both analog and digital are called "Analog Mixed Signal" or abbreviated as AMS. In the most basic block diagram of a PLL (Figure 1), the building blocks of the PLL are identified. The voltage-controlled oscillator (or VCO), the charge pump (or loop amplifier), and the loop filter are all analog blocks. The phase detector and dividers are digital blocks. Because the PLL is composed of both analog and digital blocks, it is called mixed signal. The PLL is a feedback loop that adjusts the phase and frequency of the VCO to lock to the phase of the input reference oscillator. When the PLL is locked, the output frequency is a fractional multiple of the input frequency (see eq. 1.0).

$$F_{out} / N = F_{ref} / R \quad \text{or} \quad F_{out} = F_{ref}\ N / R \qquad \text{eq. 1.0}$$

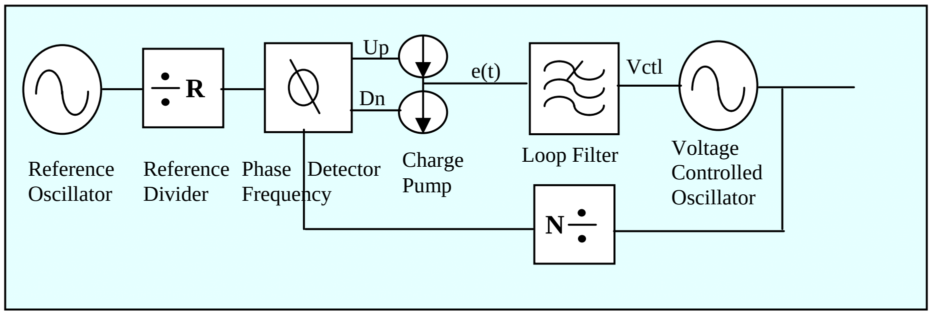


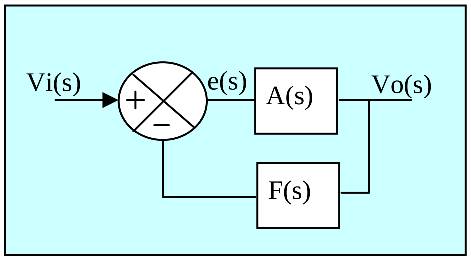
Figure 1. The basic block diagram of an integer PLL.

SIMULATION OF PLLs

Traditionally, mixed signal blocks like a PLL have been modeled with an analog simulator like SPICE (the ubiquitous IC simulation tool developed by UC Berkeley in the 1970s). However, analog tools like SPICE are relatively slow and are not readily integrated with digital simulation tools like Verilog. Verilog was developed in the 1980s for the purpose of simulating very-large-scale digital systems with moderate computing resources. Recently, mixed signal tools like Verilog-AMS have been developed to co-simulate both the digital and analog blocks simultaneously. However, these tools are usually only offered by large EDA companies with a large price tag and a steep learning curve. We will show how it is possible to model the PLL with a pure digital simulation tool like Verilog. There are a number of affordable commercial Verilog simulators and a growing number of open source simulators offered **[1]**. Using a pure digital tool like Verilog, not all of the PLL physics can be modeled. Specifically, phase noise cannot be modeled with Verilog efficiently. There is also an open source simulator tools like "cppsim" from MIT **[2]** which are specifically targeted at phase-locked loops. Cppsim offers mixed signal simulation capabilities, including phase noise simulation. Commercial tools like Agilent technologies' "Advanced Design System" **[3]** can efficiently model PLLs, including noise with technologies like Harmonic Balance and Envelope simulation .

LINEAR FEEDBACK LOOP THEORY

Because the PLL is a feedback loop, it must be designed carefully to be stable and have a well behaved closed-loop performance. A feedback loop (or control loop) is a universal building block for electronic systems **[4]**. The basic topology of a feedback loop is given in Figure 2. A pure linear feedback loop can be represented in either frequency domain or time domain. For the purposes of stability analysis, we will look at the frequency domain characteristics. The variable "s" corresponds to the response to the Laplace domain representation. Tools like Matlab or the open source equivalent Octave can be used to analyze the time domain response or the Laplace domain response of the feedback loop. The closed-loop frequency response of the PLL can be computed in the Laplace representation with s=exp(jwt) using eq. 1.5. The open loop gain is indicated in eq. 1.4.



$$Vo = A(s) * e(s) \qquad \text{eqn 1.2}$$

$$Vi(s) - F(s)Vo(s) = e(s) \qquad \text{eqn 1.3}$$

$$OLG(s) = A(s) \cdot F(s) \qquad \text{eqn 1.4}$$

$$\frac{Vo(s)}{Vi(s)} = \frac{A(s)}{1 + A(s)F(s)} \qquad \text{eqn 1.5}$$

Figure 2. Linear feedback system block diagram.

For purposes of analyzing stability feedback loops, the open-loop gain transfer function can be examined. If the phase of the open-loop gain is greater than 180 degrees when the magnitude crosses 0 dB, then the loop will be unstable. The phase at the 0 dB open-loop gain – 180 degrees is known as the phase margin (see Figure 3). For good loop stability, phase margins should be greater than 45 degrees **[4]**. This is an important factor when designing PLL loop filters and modeling the loop performance, as we will discuss later.

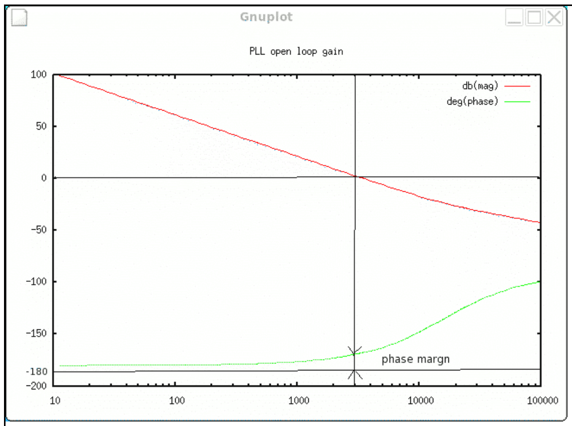
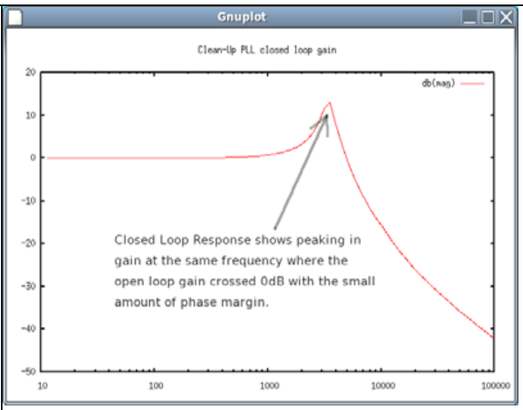


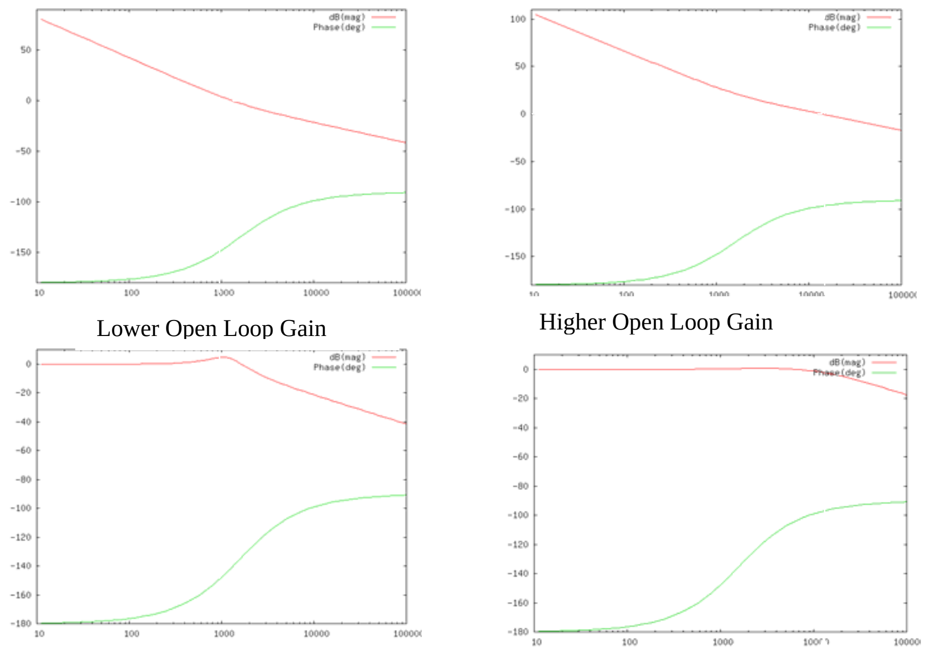
Figure 3. Open-loop phase gain of a stable PLL (plotted using Gnu Octave).



Closed Loop Response shows peaking in gain at the same frequency where the open loop gain crossed 0dB with the small amount of phase margin.

Figure 4. Closed loop response of an almost unstable PLL.

The small signal design of the loop filter is very important for PLL applications. The closed-loop response of the PLL resembles a low pass filter. By using a low loop bandwidth, the PLL can be used to filter out spurious signals from the incomming reference. By using a high loop bandwidth, a faster loop acquisition time can be obtained. Also, the PLL will track the noise of the reference within the loop bandwidth. This is commonly used to suppress the VCO noise in a monolythic PLL where the VCO resonator Q is low and the VCO phase noise is high. The closed-loop bandwidth of a PLL is approximately equal to the point where the open-loop gain crosses 0 dB. This relationship between open-loop gain and closed-loop bandwidth is shown graphically in Figure 5 below.



Lower Open Loop Gain

Higher Open Loop Gain

Closed Loop BW

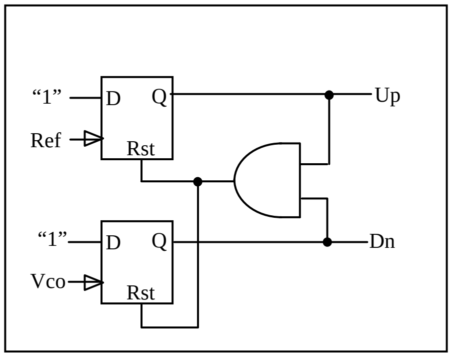
Closed Loop BW

Figure 5. The impact of open-loop gain on closed-loop bandwidth.

Next we will look at the building blocks of the PLL and show how they are designed and modeled to make a well designed PLL.

THE PHASE DETECTOR MODELED IN VERILOG

The heart of the PLL is the phase detector. The phase detector compares the phase of the reference to the VCO phase. There are many types of phase detectors; the design considerations include: noise, phase capture range, and frequency capture range. A comparison of phase detectors is given in **[3]**. The schematic of a popular phase-frequency detector is shown in Figure 6. If the VCO phase lags the reference, then the phase detector produces an UP pulse. If the VCO phase is ahead of the reference, it produces a DOWN pulse. The digital output of a phase frequency detector is shown in Figure 7. The up/down pulses go to a loop amplifier or charge pump, where they get converted into an analog electrical signal. Sometimes, instead of a charge pump, the phase detector has a tri-state output that can drive a opamp loop filter directly. This signal is conditioned by the charge pump and loop filter which drives the control voltage (Vcontrol) of the VCO.



```
assign ff_rst = dn & up;     //  the AND gate

always @( posedge ref   or posedge ff_rst or negedge rstN )
begin
   if (!rstN) up<=1'b0;
   else if (ff_rst) up <= 1'b0;
               else up <= 1'b1;
end
always @( posedge vcxo   or posedge ff_rst or negedge rstN )
begin
   if (!rstN) dn<=1'b0;
   else if (ff_rst) dn <= 1'b0;
               else dn <= 1'b1;
end
```

Figure 5. A typical phase-frequency detector.

Verilog 1: The verilog for the phase-frequency detector.

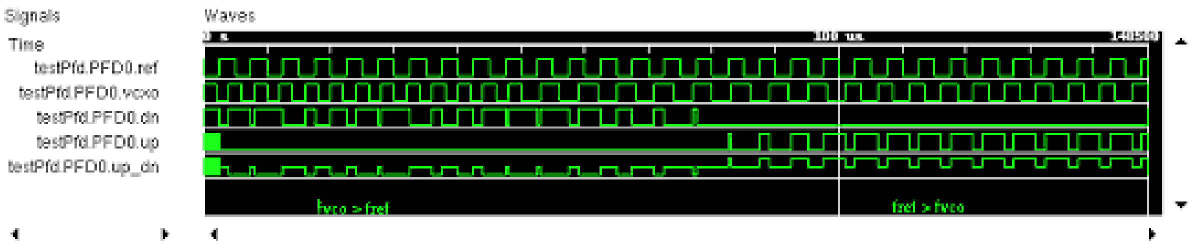


Figure 6. Up Dn pulses and a tri-state output of a phase-frequency detector.

After filtering with the charge pump and loop filter, the pulses are converted into a DC voltage. This voltage is plotted as a function of phase shift in the Figure 8. The periodic nature of the phase detector results in a nonlinear sawtooth-like response, however, in the range of $-\pi < \theta < \pi$.

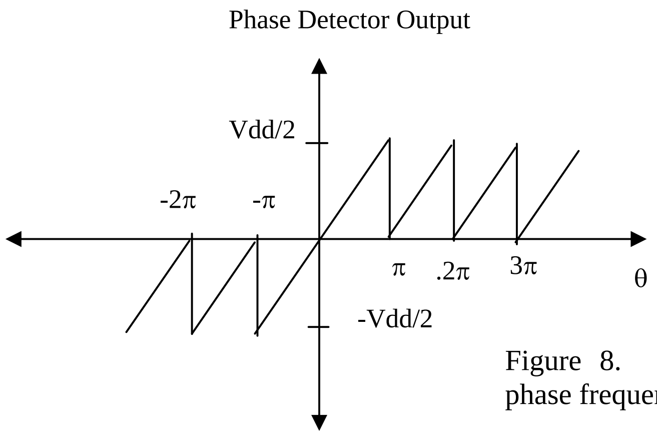


Figure 8. The Transfer curves for the phase frequency detector.

The gain of the phase detector is linear with gain Kϕ=Vdd/2π **[7]**. When the phase error exceeds the range of ($-\pi < \theta < \pi$), it is referred as a "cycle slip" **[6]**. For very large frequency errors, the phase error can be very large, which has the tendency to lower the effective phase detector gain and slow down the acquisition time. Some vendors offer the ability to boost the loop gain during "cycle slips" and speed up the acquisition time. We will demonstrate the closed-loop locking impact of cycle slips later on in this paper.

THE VCO MODELED IN VERILOG

The VCO takes a control voltage input a and creates a frequency porportional ot the control voltage. The phase of the VCO is the integral of the frequency (eq. 1.6). Because of this, the Laplace transform of the phase has a (1/s) or (1/jw) (eqn 1.7). This results in 90-degree phase shift. The best phase margin that can be expected from a PLL is 90 degrees.

$$f_{\mathrm{vco}} = f_c + k_v \cdot V_{ctl} \qquad \phi_{vco}(t) = \int 2\pi \cdot (f_o + k_v \cdot V_{ctl}(t))dt \qquad \text{eqn 1.6}$$

$$\phi_{vco}(s) = k_v \cdot V_{ctl}(s)/s \qquad \text{eqn 1.7}$$

The VCO can be modeled in Verilog using the "always" command. Verilog clocks are modeled as an inversion of the signal after a delay of Period/2, as shown below. The "always" statement causes the phase to be repetitively inverted after the delay given after the "#" symbol. This generates a square-wave signal that has the frequency of 2/Period;

```
always #(Period/2)  vco <= ~vco;
```

The extension to make this look like a VCO is straightforward. VCOs are frequency-controlled devices, not delay- or period-controlled devices, but since period is the reciprocal of frequency, we can derive one from the other (eq. 1.8).

$$Period = \frac{1}{f_o + k_v \cdot V_{ctl}} \quad Period \approx \frac{1}{f_o} \cdot \left(1 - \frac{k_v \cdot V_{ctl}}{f_o}\right) \qquad \text{eqs. 1.8, 1.9}$$

The VCO is accomplished with the following Verilog statement:

```
always #((1/Fo – Vctl * Kv/Fo^2)/2)   vco <= ~vco;
```

Verilog 2: The VCO modeled in Verilog as a clock with variable period.

Care must be taken because all Verilog simulators do not support an event model that allows real variables in the #delay statement. Some careful simulation of this statement may be needed to make sure that the simulator works correctly.

THE LOOP FILTER MODELED IN VERILOG

If the phase detector is has a tri-state output, then it can directly drive an opamp based loop filter. Most loop filters are based upon an integrator loop. The integrator loop filter is advantageous, because it has zero steady-state phase error, but it adds another 1/s or 90 degrees to the open-loop gain. This results in a total of 180 degrees of phase shift, which will cause positive feedback and oscillations. This is circumvented by the use of a lead network. This loop filter looks like Figure 9.

$$H(s) = \frac{1 + s \cdot Rf \cdot Cf}{s \cdot Ri \cdot Cf} \qquad \text{eqn 2.0}$$

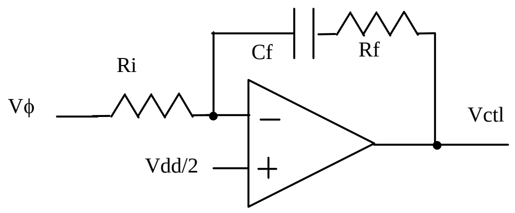


Figure 9. A PI analog loop filter made with an operational amplifier.

Because the loop filter is an analog quantity, it can not be simulated using Verilog. However, there is a digital equivalent to the loop filter used in sampled data control systems. It is referred to as the PID filter (Proportional, Integral, Derivative). In our loop filter, we do not use derivative information, so, technically speaking, it is a PI filter. The integrator of the filter is just replaced with an accumulator. The opamp circuit can be replaced by a digital filter using Z-transform theory z=exp(jwT), where T is the sampling period. The difference approximation for differentiation is s=(1-z^{-1})/T,. It follows that the integrator is 1/s=T/(1-z^{-1}). The PI digital filter will just be an integrator (implemented with an accumulator or adder) and summed with a proportional term. Note that the sample period must be smaller than the reciprocal of the closed-loop bandwidth, so that the loop control voltage meets the Nyquist sampling criteria. Ideally, the digital loop filter sample clock should be an order of magnitude faster than the PLL Reference input.

It is worth noting that newer generation PLLs often implement the loop filter digitally in either the FPGA, a DSP processor, or as an algorithm in a micro controller. This digital loop filter drives an ADC, which generates the Vctl voltage for the VCO. These digital PLL loop filters are much easier to model in a digital language like Verilog than in an analog simulation language like SPICE. These digital loop filters can be transformed into an analog equivalent for the purpose of doing the Nyquist stability analysis.

```
always @ (posedge smplCk ) begin
     if     (PdUp==1'b1)  begin
      integral = integral + FILT_I;
      porportional  = FILT_P;
end
else if (PdUp==1'b0)  begin
       integral = integral - FILT_I;
       porportional = - FILT_P;
end
else begin
       porportional=0.0;
end
Vctl = integral + porportional;
End

always #((To – Vctl * Kphi )/2)   vco <= ~vco;
```

$$P = \frac{Rf}{Ri} \qquad \text{eqn 2.1}$$

$$I = \frac{1}{Ri \cdot C_f} \qquad \text{eqn 2.2}$$

$$V_{ctl} = V\phi \cdot (P + \frac{I \cdot T}{1 - z^{-1}}) \qquad \text{eqn 2.3}$$

Verilog 3: The Verilog code for the Loop Filter and the relationship with the analog loop filter.

VERILOG SIMULATION RESULTS

The following blocks are assembled in a Verilog testbench to examine settling time and loop stability using a Verilog simulator. First, we simulate an underdamped situation where the proportional term of the PI loop filter is set near zero. The Verilog testbench changes the frequency of the reference at 10000 ns to 10.2 MHz and then changes the reference to 9.8 MHz at 50000 ns. The PLL attempts to track the changes, but oscillates because the phase margin is near zero. This is shown in Figure 10 below. In contrast, the proportional term is set to a critically damped value. When a loop is critically damped, the loop tracking has optimal speed. These results are shown in Figure 11 below.

Other PLL transient effects can be simulated with this Verilog testbench. Recall the phase frequency detector curve from Figure 8 above. When the phase error is a multiple of π, the output of the phase frequency detector goes to zero. This causes the slew rate of the frequency vs. time to go to zero and slows the locking of the loop to a frequency change. The result of cycle slips in PLL transient response is shown in Figure 12. Phase slips occur at 3000 ns, 6000 ns, and 9000 ns.

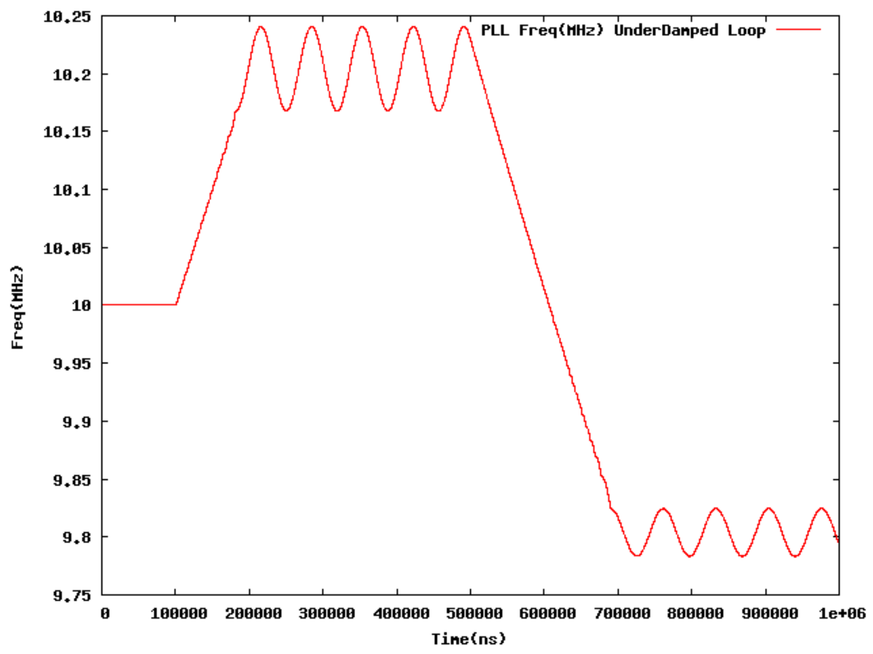


Figure 10. Verilog simulation of underdamped PLL.

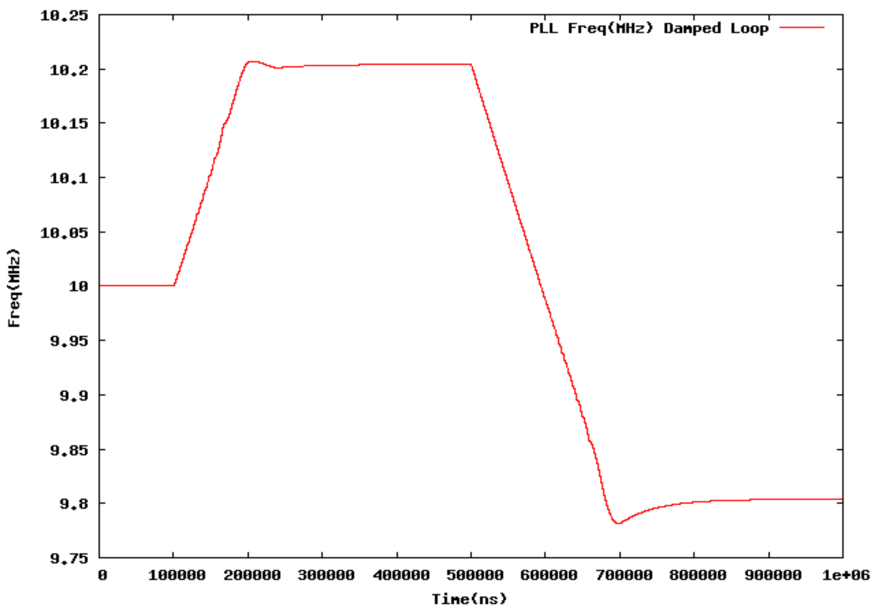


Figure 11. Verilog simulation of critically damped PLL.

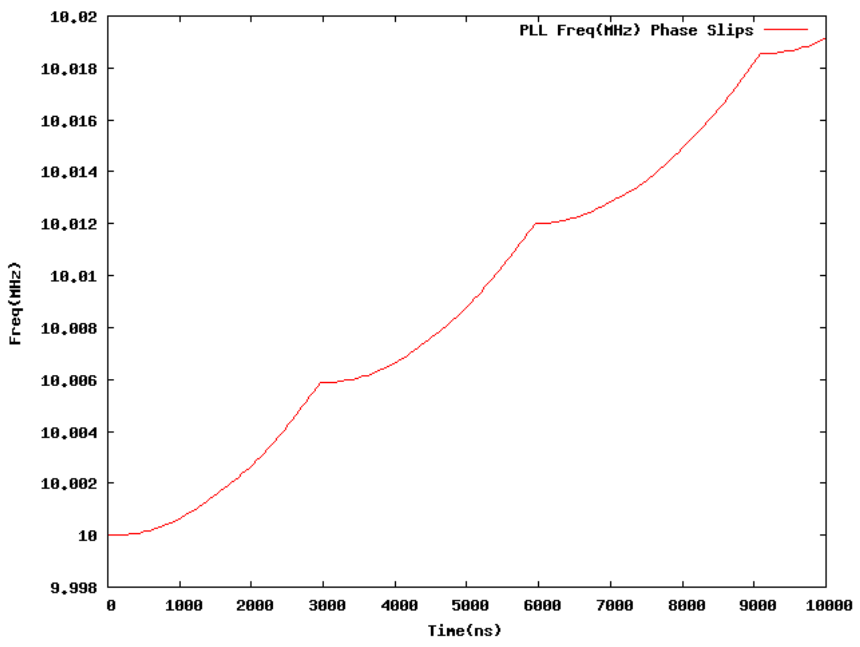


Figure 12. Verilog simulation of cycle slips.

CONCLUSIONS

It is possible to use a strictly digital tool like Verilog to simulate the transient characteristics of a phase-locked loop. The analog portions of the PLL must be modeled with a digital filter, since Verilog has no analog modeling capability. The impact of loop filter design on loop stability in transient behavior can be simulated. Moreover, complicated digital circuitry like sigma-delta modulated fractional-N control logic can be modeled easily in Verilog with its digital centric capabilities. However, not all characteristics of a PLL can be modeled in Verilog. Specifically, the PLL phase noise cannot be modeled. To simulate noise, more specialized PLL tools like MIT's "cppsim" can be used **[2]**, or commercial tools like Agilent's "Advanced Design System" can be purchased **[3]**.

REFERENCES

[1] Free Verilog Tools, *http://www.verilog.net/free.html*

[2] M. H. Perrott, 2002, "*CppSim Reference Manual,*" (MIT High Speed Circuits and Systems Group).

[3] Agilent Technologies, 2006, *"Advanced Design System Documentation"* (Agilent Technologies, Santa Clara, California).

[4] R. Dorf and R. Bishop, 1998, **Modern Control Systems** (Prentice Hall, Englewood Cliffs, New Jersey), ISBN 0-201-30864-9.

[5] G. F. Franklin, J. D. Powell, and M. L. Workman, 1997, **Digital Control of Dynamic Systems** (3rd edition) (Addison-Wesley, Reading, Massachusetts), ISBN 0-201-82054-4.

[6] D. Banerjee, 2006, **PLL Performance, Simulation, and Design** (4th edition), National Semiconductor, Inc., http://www.national.com/appinfo/wireless/files/deansbook4.pdf

[7] R. E. Best, 2003, **Phase-Locked Loops: Design, Simulation, and Applications** (McGraw-Hill, New York), ISBN 0-07-141201-8.