

# Behavioral Modeling of PLL Using Verilog-A

## Introduction

In this article, we describe practical behavioral modeling for highly non-linear circuits using Verilog-A, which is analog extension of Verilog-AMS. At first, we describe behavioral modeling techniques for phase/frequency detectors (PFD) and voltage-controlled oscillators (VCO) those are essential part of phase-locked loop systems shown in Figure.1. Model parameter extraction techniques are described and demonstrated later. Finally, these models are simulated with SmartSpice and verified against the results of transistor circuit simulations.

## Phase Detector (PD)

A well-known sequential-logic PFD shown in Figure.2(a) is used for phase/frequency detection circuit. This PFD produces UP and DOWN signals depending on the phase difference between reference clock and feedback clock (Figure.2(b)).

## Phase Detector (PD)

A well-known sequential-logic PFD shown in Figure.2(a) is used for phase/frequency detection circuit. This PFD produces UP and DOWN signals depending on the phase difference between reference clock and feedback clock (Figure.2(b)).

The behavioral model of PFD can be represented as shown in List.1. This module monitors the phase difference between the clocks. When the rising edge of

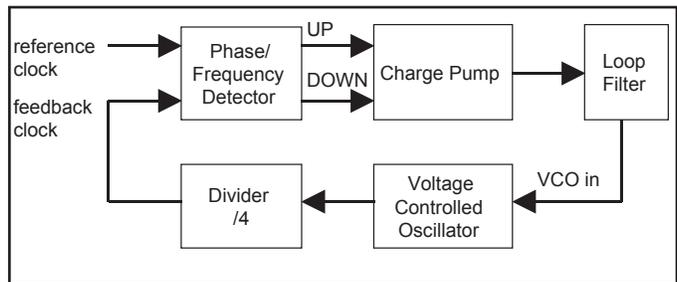


Figure 1. PLL Block Diagram.

REF clock occurs before the rising edge of FB clock, the module activate UP signal and then forces the VCO to increase the oscillation frequency. In the reverse case, the module activate DOWN signal and forces the VCO to decrease the frequency. Note that UP and DOWN signal are active-low and active-high respectively.

## List 1. Behavioral representation of PFD in Verilog-A

```

module pll_pd (ref_clk, fb_clk, up_out, down_out);
  inout  ref_clk, fb_clk, up_out, down_out;
  electrical ref_clk, fb_clk, up_out, down_out;
  parameter real vdd=3.3,
               ttol=10f,
               ttime=0.2n ;
  integer state; // state=1 for down, -1 for up
  real td_up, td_down ;

```

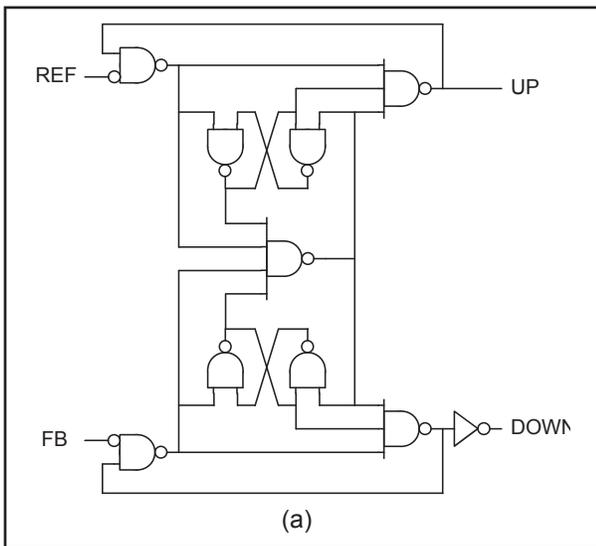


Figure 2a. PFD circuit.

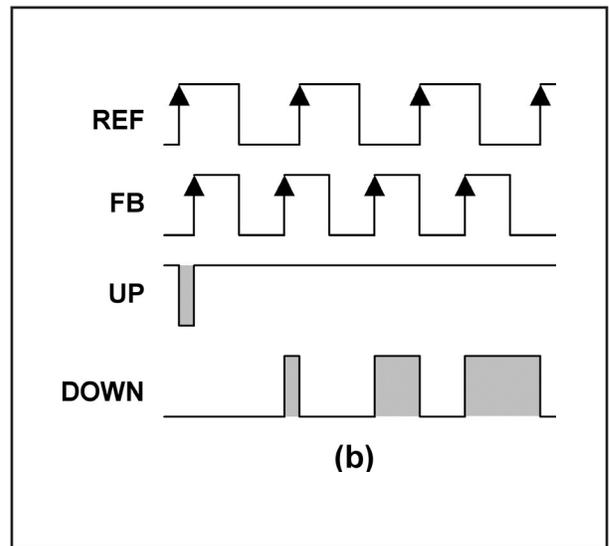


Figure 2b. Basic operation.

```

analog begin
  @(cross( V(ref_clk) - vdd/2 , 1 , ttol )) begin
    state = state - 1;
    if(V(up_out)>vdd/2) td_up=480p; else td_up=1005p;
    if(V(down_out)<vdd/2)td_down=480p; else td_down=1090p;
  end
  @(cross( V(fb_clk) - vdd/2 , 1 , ttol )) begin
    state = state + 1;
    if(V(up_out)>vdd/2) td_up=480p; else td_up=1005p;
    if(V(down_out)<vdd/2) td_down=480p; else td_down=1090p;
  end

  if ( state > 1 ) state = 1 ;
  if ( state < -1 ) state = -1;
  V(down_out) <+ transition( (state + 1)/2*vdd , td_down , ttime );
  V(up_out) <+ transition( (state - 1)/2*vdd+vdd , td_up , ttime );
endmodule

```

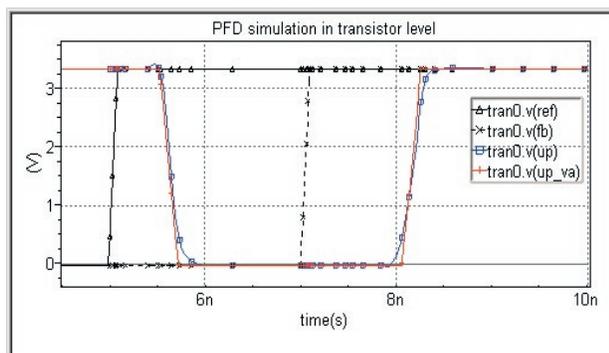
In this module, we use td\_up and td\_down variables to define delay time of input-to-UP and input-to-DOWN respectively. Since delay time of rise transition is different from fall transition, different values should be assigned to these variables depending on output transition direction. The following codes used in the module allow designers to define different delay time for rise and fall transition independently.

```

if(V(up_out)>vdd/2) td_up=480p; else td_up=1005p;
if(V(down_out)<vdd/2) td_down=480p; else td_down=1090p;

```

The optimal value of variables can be obtained from transistor level simulation shown in Figure.3 (a) and (b). From these plots, we can obtain the optimal value of td\_up=480ps when rising and td\_up=1005ps when falling, and td\_down=480ps when rising and td\_down=1090ps when falling. By using these optimal values, we can see good agreement in the output waveforms of both transistor level and Verilog-A simulations.



## Voltage Controlled Oscillator (VCO)

An ideal voltage-controlled oscillator generates a periodic output signal whose frequency is a linear function of control voltage. The output frequency  $f_{out}$  can be expressed as;

$$f_{out} = f_{nom} + gain \cdot V_{cont}$$

where  $f_{nom}$  is a nominal (or free running) frequency, gain is the gain of VCO in Hz/V, and  $V_{cont}$  is a control voltage supplied from charge pump. Since phase is the time-integral of frequency, the sinusoidal output of VCO can be expressed as;

$$V_{out}(t) = A \cdot \sin(2 \cdot \pi \cdot \text{integral}(out)) + V_{offset}$$

where  $A$  is a amplitude of sinusoidal and  $V_{offset}$  is an output offset voltage.

With based on the above principle, behavioral model of VCO can be represented as shown in List.2.

### List 2. Behavioral representation of VCO in Verilog-A

```

module pll_vco ( in , out ) ;
  inout   in , out ;
  electrical in , out ;

  parameter real vdd = 3.3 , // operational voltage
              amp = vdd/2 , // amplitude of vout
              offset = vdd/2 , // offset of vout
              gain = 540e6 , // gain [Hz/V]
              vnom = 1.27 , // nominal vin
              fnom = 400e6 ; // frequency at vnom

```

```

  real   freq ;

```

```

  analog begin

```

```

    freq = fnom + gain*(V(in) - vnom) ;

```

```

    V(out) <+ amp*sin(2*M_PI*idt(freq)) + offset ;

```

```

  end

```

```

endmodule

```

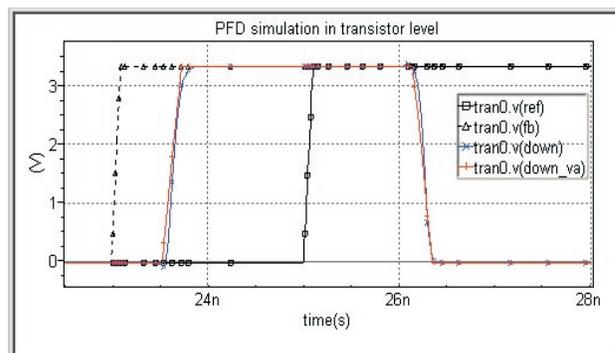


Figure 3. Simulation results of PFD. Shown are (a) UP waveforms, and (b) DOWN waveforms.

**Note** that,  $V_{cont}$  is replaced by  $V_{(in)} - v_{nom}$  in the module description.

The module parameters from is the nominal frequency and is defined in design specification. The 'offset' may be defined as half of  $V_{DD}$ . The  $v_{nom}$  and gain should be extracted from transistor level simulation. Figure.4 shows the transistor level simulation result of VCO frequency respect to control voltage.  $v_{nom}$  can be found by measuring control voltage at which VCO frequency is equal to 400MHz, and gain can be obtained by calculating the gradient of the curve at  $V_{cont} = v_{nom}$ . In this example, we obtained  $v_{nom} = 1.27v$  and  $gain = 540MHz/V$  from the plot.

Verilog-A simulation result is also displayed in Figure.4. Since we assume the VCO gain is a linear function of control voltage, there's a discrepancy in low and high regime of  $V_{cont}$ . However, you can overcome this issue by adapting non-linear function for gain expression.

## Conclusion

By using behavioral modeling technique, and a mixed signal simulation with SmartSpice, PLL system can be efficiently simulated with good accuracy and reasonable run-time cost. The module characteristic can be parameterized in behavioral representation, and the parameters can be easily extracted from the transistor level simulation. A designer would thus start the design process by using an all behavioral model of PLL and focusing on optimizing module parameters. Once enough performance is obtained, the modules can be widely used for various subsystems and in a mixed signal simulations in future designs.

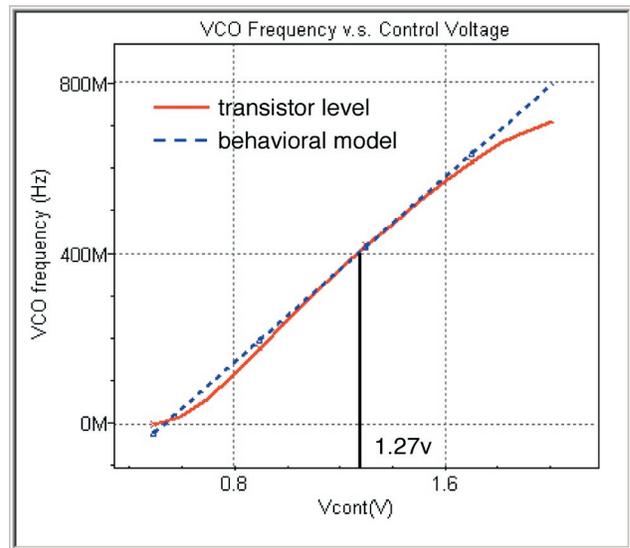


Figure 4. Simulation results of VCO.