Fast Mixed-Mode PLL Simulation Using Behavioral Baseband Models of Voltage-Controlled Oscillators and Frequency Dividers

Ihor Harasymiv, Manfred Dietrich, Uwe Knöchel
Design Automation Division
Fraunhofer Institute for Integrated Circuits
Dresden, Germany
{ihor.harasymiv, manfred.dietrich, uwe.knoechel}@eas.iis.fraunhofer.de

***Abstract*— This article presents a new approach to fast mixed-mode simulation of phase-locked loops (PLLs) in time domain using Spice-like simulators and behavioral Verilog-A baseband (BB) models of voltage-controlled oscillators (VCO) and frequency dividers (FD). Other PLL blocks like phase-frequency detectors (PFD), charge pumps (CP), and loop filters (LP) can be transistor level and/or behavioral models. The use of both VCO and FD BB models in mixed-mode test bench allows fast PLL simulation and optimization of modern sophisticated PFD and CP blocks on transistor level with speedups of about 2-3 orders of magnitude.**

Keywords- phase-locked loop; N-integer frequency synthesizer; passband; baseband; signal; behavioral model; Verilog-A; Spice-like simulator

I. INTRODUCTION

Phase-locked loops (PLLs) are widely used in modern analog, digital and mixed-signal electronic systems [1] for signal synchronization, jitter reduction, skew suppression, clock generation and recovery, and frequency synthesis (Fig.1). PLLs are strong nonlinear feed-back systems that exhibit complex dynamics:

- Several locked states - periodic and/or almost periodic steady-states,
- Complex behavior in the unlocked state,
- Large acquisition (settling) time (i.e., long transition from one steady-state to another) - 4-6 orders of magnitude greater than the period of the voltage-controlled oscillator (VCO) ,
- Large bandwidth: Reference frequency is typically 2-4 orders of magnitude lower then the VCO frequency.

Therefore, traditional simulation of PLL on transistor level using Spice-like simulators requires significant CPU time. Behavioral models (see, e.g., Verilog-A and Verilog-AMS [2] models in [3-5]), can reduce the CPU time to some degree, but they can not eliminate the inefficiency of transient simulations due to the "large bandwidth" and "two time scales" [5] problems. Thus, several methods and techniques for fast PLL simulation are proposed in literature.

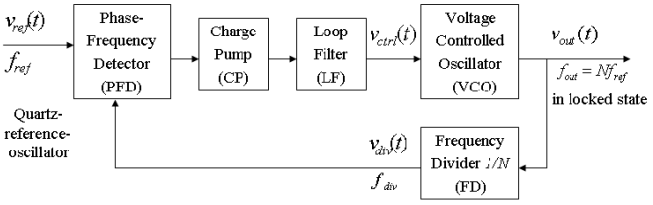

Figure 1. PLL example: Block diagram of N-integer frequency synthesizer.

For example, reference [6] proposed SPICE-compatible phase-space simulation techniques based on well known analytic linear phase domain (PD) models. Transition to PD (phases vs. time) eliminates the "two time scale" problem, resulting in speedups of 2-3 orders of magnitude. However, this speed-up is obtained at the expense of accuracy, because simple PD macro-models of PFD and VCO can not model all effects in transistor level blocks accurately. Better modeling properties can be obtained with nonlinear VCO phase models as proposed in [7] for small control voltages as well as in [8] for large voltages using trajectory piecewise techniques. They are based on "perturbation projection vector" (PPV) techniques, that capture nonlinear effects in VCO accurately while providing PLL simulation speed-ups of more than two orders of magnitude.

A time-domain (TD) method for fast PLL/DLL simulation [9] is based on uniform time sampling and two techniques for accurate discrete time representation of continuous time PFDs and VCOs/FDs respectively. Discrete PFD output is created from a continuous one using an area conservation principle. The simulation sample frequency is considerably reduced due to inclusion of the FD model into VCO simulation module.

A mixed PD-TD method of fast PLL simulation [10] extends the envelope-transient methods of nonlinear circuit analysis (see, e.g., overviews in [10-11]) to the PLL analysis. In this method, all PLL signals are expressed as Fourier series expansions with unknown time varying complex coefficients $C_k(t)$,

$$x(t) = \sum_{k=-NH}^{NH} C_k(t) \cdot e^{j\omega_k t} , \qquad (1)$$

978-1-4244-6815-7/10/$26.00 ©2010 IEEE

where *NH* is the number of known frequencies ω_k (i.e., carrier frequencies, their combinations and harmonics). The number of equations to be solved is *2·NH+1* times greater than the number of unknown PLL signals. Therefore, it is reasonable to investigate the efficiency and accuracy of mixed-mode PLL simulation, when (1) is applied only with a single carrier frequency – i.e., when *NH = 1*. Moreover, for PLL simulation it is reasonable to apply (1) only to the VCO output signal, which is also the input signal of the FD.

II. BASEBAND MODELING OF VCOS AND FDS

The baseband (BB) modeling approach is well known and widely used for analysis of communication systems and linear and nonlinear circuits [12]. BB models are efficient not only for analytic analysis of linear time invariant (LTI) systems, but also for numerical simulation of nonlinear HF/RF blocks and sub-systems using Spice-like simulators [13-14]. The TD passband (PB) signal *x(t)* can be written as

$$x(t) - x_0 = A(t)\cdot\cos(\omega_0 t+\varphi(t)) = I(t)\cdot\cos(\omega_0 t) - Q(t)\cdot\sin(\omega_0 t), \quad (2)$$

where $I(t)= A(t)\cdot\cos(\varphi(t))$ and $Q(t)=A(t)\cdot\sin(\varphi(t))$ are in-phase and quadrature parts of corresponding complex BB signal $x_{BB}(t)=I(t)+j\cdot Q(t)$ respectively, $\omega_0=2\pi\cdot f_0$ is the carrier frequency, *A(t)* the amplitude, $\varphi(t)$ the phase deviation, and x_0 is constant.

Note 1: The PB signal representation (2) is equivalent to a Fourier series expansion (1) in case of *NH = 1*.

Note 2: The popular statement that BB models increase the speed of numerical simulation because they "suppress the carrier frequency" is not entirely correct: The carrier frequency in BB models is not suppressed – it is taken into account analytically. The BB model of a LTI system is also linear [14]:

$$Y_{BB}(s)=H_{PB}(s+j\cdot\omega_0)\cdot X_{BB}(s), \quad (3)$$

where $X_{BB}(s)$ and $Y_{BB}(s)$ are Laplace transforms of the input $x_{BB}(t)$ and the output $y_{BB}(t)$ respectively, $H_{PB}(s)$ is the Laplace transfer function of the PB LTI.

Moreover, reference [14] proposed and tested a signal flow BB model of nonlinear HF/RF blocks and sub-systems such as low noise (LNA) and power amplifiers (PA), modulators, demodulators, mixers, and direct conversion receivers (Fig.2).

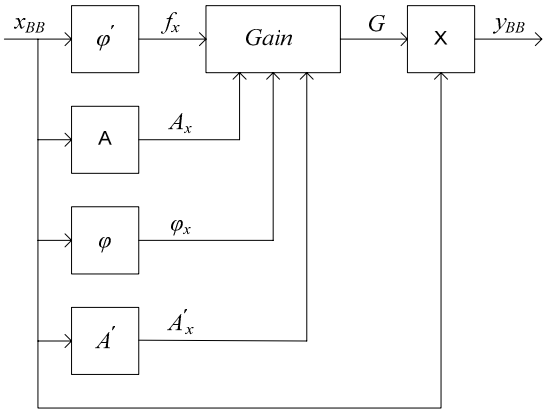

Figure 2. Block diagram of signal flow baseband model

The four input functions of the general BB model determine the instantaneous frequency $f_x(t)$ as time derivative of the phase $\varphi'_x(t)$, the amplitude $A_x(t)$, the phase $\varphi_x(t)$, and the time derivative of the amplitude $A'_x(t)$ of the BB input signal [14]. Different HF/RF blocks require a different number of input functions. For example all frequency-dependent RF blocks (e.g., filter) contain a $\varphi'_x(t)$ input function. Consequently, for LTI blocks no further input functions are necessary, because the complex gain only depends on the frequency. The BB models of nonlinear blocks (LNA, mixer) additionally contain the input function $A_x(t)$. Therefore AM/AM and AM/PM curves, representing the 1dB compression and IP3 points, can be modeled in the complex BB. The input function $\varphi_x(t)$ must be used for all time-variant RF blocks such as mixer, IQ-modulator and IQ-demodulator. PM/AM and PM/PM curves can be modeled using this input function. For the modeling of PAs the input function $A'_x(t)$ is necessary.

The complex BB gain $G =(I_y+j\cdot Q_y)/(I_x+j\cdot Q_x)= ReG+j\cdot ImG$ is calculated during nested swept periodic steady-state analyses (PSS) for the input amplitude A_x, frequency f_x and phase φ_x. The calculated values are stored in two multidimensional tables, one for the real part and one for the imaginary part. The tables can be imported with the Verilog-A function `$table_model()`. Interpolations and extrapolations are handled by this function of one or several variables. The degree of the spline interpolation (1-3), as well as the type of extrapolation (clamp, linear, spline and error) can be defined via function parameters.

In this paper we propose to create BB models of VCOs and FDs using the following expression [14] for the instantaneous frequency:

$$f(t)=f_0+\varphi'(t)/2\pi, \quad (4)$$

where $\varphi'(t)=d\varphi/dt=(I(t)\cdot Q'(t)-Q(t)\cdot I'(t))/(I^2(t)+Q^2(t))$. (5)

More precisely, both VCO and FD models are "semi-baseband" models: The baseband representation (2) has only the output signal of the VCO and the input signal of FD.

A. BB Model of a VCO

Without loss of generality, we assume that the amplitude of the PB VCO output voltage is constant and the frequency f_{vco} depends only on the control voltage v_{ctrl} : $f_{vco}=F(v_{ctrl})$. This nonlinear quasi-static characteristic is calculated during a swept PSS analysis, as shown in Fig. 3, and the values of v_{ctrl} and f_{vco} are stored in one table.

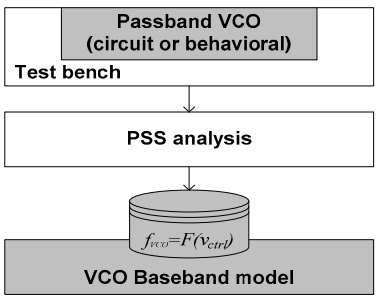

Figure 3. Baseband VCO modeling flow

The corresponding text file "VCO.tbl" has the following layout:

```
# "VCO.tbl": table model file
# for veriloga view of cell vcoBB
# vctrl (V)  vco_freq (Hz)
1.125   2.4914e+9
…
1.5     2.4688e+9
…
2.875   2.2793e+9
```

The Veriloga-A function `$table_model()` imports the data from file "VCO.tbl" and implements both interpolation and extrapolation of the characteristic $f_{vco}=F(v_{ctrl})$. Any frequency value from this file can be used as the model parameter "carrier frequency" f_0 in both VCO and FD BB models, but it is reasonable to choose f_0 in the middle of the VCO frequency band.

Thus, the phase deviation of the VCO output signal can be written as

$$\varphi(t)=2\pi\cdot\int_0 (F(v_{ctrl}(\tau))- f_0)d\tau\,. \qquad (6)$$

To avoid large phase values in the VCO BB model, the integral in (6) is implemented as the Verilog-A operator `idtmod()`:

```
module vcoBB(ctrl, Iout, Qout);
input ctrl; output Iout, Qout;
voltage ctrl, Iout, Qout;

parameter real fvco0 = 2.4688G; // VCO frequency
parameter real Vctrl0 = 1.5;    // at Vctrl0
parameter real Amp = 1.5;       // VCO ampl.=vdd/2
real delta_f, delta_phi;

analog begin

delta_f=$table_model(V(ctrl),"VCO.tbl","1CC")-fvco0;
delta_phi= `M_TWO_PI * idtmod(delta_f, 0.0, 1.0);
if (abs(delta_f)!= 0)
$bound_step((1.0/abs(delta_f))/32);

V(Iout) <+ Amp * cos(delta_phi);
V(Qout) <+ Amp * sin(delta_phi);

end
endmodule // vcoBB
```

The proposed BB model of the VCO can be easy modified if the frequency and the amplitude of the VCO output voltage depend on several voltages (e.g., control, source, bias voltages). In this case, the required data files are created during the corresponding nested swept PSS analyses.

B. BB Model of a FD

The behavioral BB model of the FD can be created using its PB transistor level or behavioral model. For example, in our test bench (TB) PLL_PB (Fig.4) we use the behavioral PB model of FD that generates output pulses with a duty ratio of 0.5. The instantaneous frequency of the FD output signal is N_{div} (divider ratio) times lower than the instantaneous frequency at its input:

$$\omega_{div}(t)=(\omega_0+\varphi'(t))\,/N_{div} = \omega_0/N_{div} +\varphi'(t)/N_{div}\,, \qquad (7)$$

where $\varphi'(t)$ is calculated from the FD input BB signal using (5). Thus, the corresponding instantaneous phase $\varphi_{div}(t)$ in the divider is equal to

$$\varphi_{div}(t) = \int_0 \omega_{div}(\tau)d\tau\,. \qquad (8)$$

The output signal of the FD BB model with a duty ratio of 0.5 is generated using the following events: phase $\varphi_{div}(t)= k\cdot\pi$ – i.e., when $sin(\varphi_{div}(t))$ crosses zero threshold in direction +1 or -1. Therefore, the corresponding Verilog-A BB model of the FD is:

```
module dividerBB(Iin, Qin, out, ctrl);
input Iin, Qin, ctrl;output out;
voltage Iin, Qin, out, ctrl, dI, dQ;

parameter real f0=1.0G;        // VCO nom. frequency
parameter integer dir = +1 from[-1:+1];
parameter real Vhigh=3.0;      // voltage VH
parameter real Vlow=0.0;       // voltage VL
parameter real trf=10n;        // rise time=fall time
parameter real tdel=10n;       // output delay
// norm. voltage to set divider ratio
parameter real Vnorm=1.0m from (1.0u:inf];
real Vdif, omega_div, omega0div, phi_div, iin, qin;
integer Ndiv, B, count;

analog begin

@(initial_step) begin
Vdif=Vhigh-Vlow; count=-1; B=1;
Ndiv= V(ctrl)/Vnorm;   omega0div= `M_TWO_PI*f0/Ndiv;
end // of initial_step

iin= 1.0u * V(Iin); qin= 1.0u * V(Qin);
V(dI) <+ ddt(iin);  V(dQ) <+ ddt(qin);
omega_div=omega0div+((iin*V(dQ)- qin*V(dI))/Ndiv)
/(iin*iin + qin*qin + 1.0f);
phi_div=idtmod(omega_div,0.0,`M_TWO_PI);

@(cross(sin(phi_div),+1)) begin
if (dir==1) B=1; else B=0;
end
@(cross(sin(phi_div),-1)) begin
if (dir==1) B=0; else B=1;
end

V(out)<+ Vlow + Vdif * transition(B,tdel,trf,trf);
end
endmodule //  dividerBB
```

III. MODEL VERIFICATION

A. Passband and baseband test benches

Fig.4 shows a PLL passband testbench - a CMOS N-integer frequency synthesizer. The PB VCO is a 3-stage ring oscillator with 3 inverters (I71 – I73) and 3 transistors (M0 – M2) used as variable capacitors. The FD is a Verilog-A model that generates output pulses with a duty ratio of 0.5. The PFD (XOR type) is implemented using 2 inverters and 2 transmission gates.

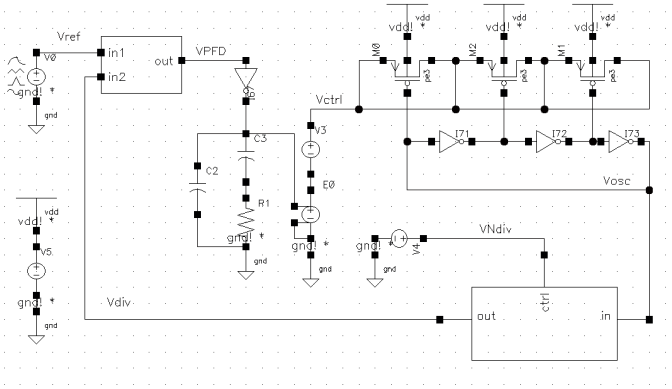

Figure 4. Schematic view of TB "PLL_PB".

Voltage source V4 sets the divider ratio, voltage sources E0 (vcvs) and V3 generate the VCO control voltage Vctrl from the output signal of the low pass filter (C2, C3, R1).

The corresponding baseband TB "PLL_BB" (Fig. 5) is created from the TB "PLL_PB" by replacing both PB models of VCO and the FD with BB ones respectively.

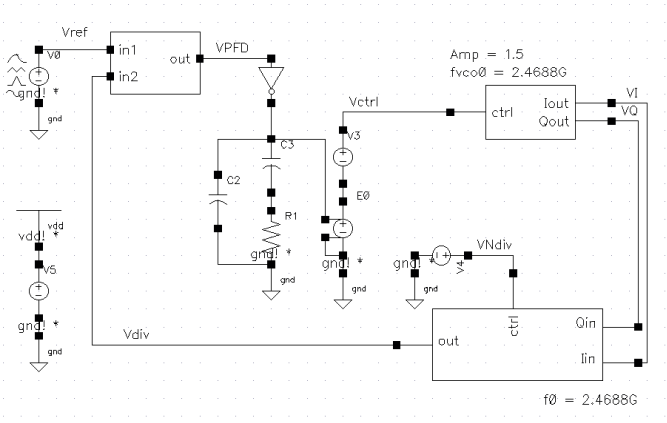

Figure 5. Schematic view of TB "PLL_BB".

Parameters in both TBs are: f_{ref}= 10.0 MHz; N_{div} = 244; i.e., the expected VCO frequency f_{VCO} is 2440.0 MHz.

B. Test benches "PLL_PB" and "PLL_BB": Comparison of simulation results

The Spectre initial transient simulation results of the PB TB are shown on Fig.6 and Fig.7. Initial conditions are: *uC2(0)= uC3(0)= 0 V*. The time interval is [0, 3.5us], and CPU time is 4270.0 s. Two instantaneous frequencies f_{div} and f_{ref} are calculated at the post-processing step, applying the Spectre calculator function *freq()* to the voltages Vdiv (FD output) and Vref (reference source) respectively. Time domain function *freq()* finds all time points t_k , where the voltage crosses the specified threshold (our settings: *1.5 V*, rising edge). Finally it calculates all values of $f_k = 1/T_k$, where $T_k = t_k - t_{k-1}$ is the k^{th} instantaneous period.

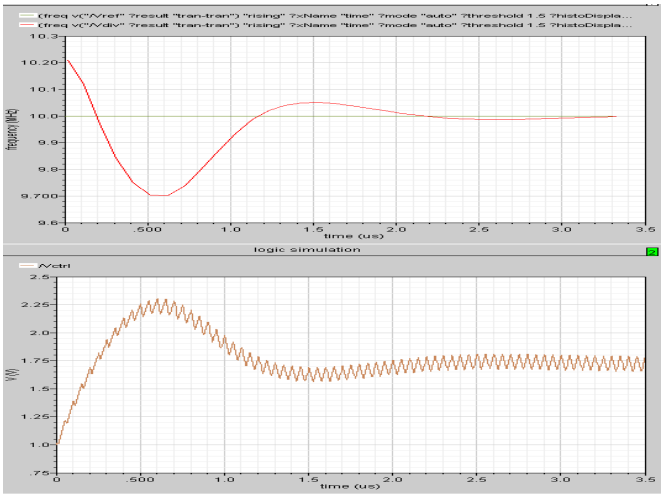

Figure 6. TB "PLL_PB". PLL initial transient response: Frequencies f_{div} and f_{ref} (top), VCO control voltage Vctrl (bottom); time interval: [0, 3.5µs] .

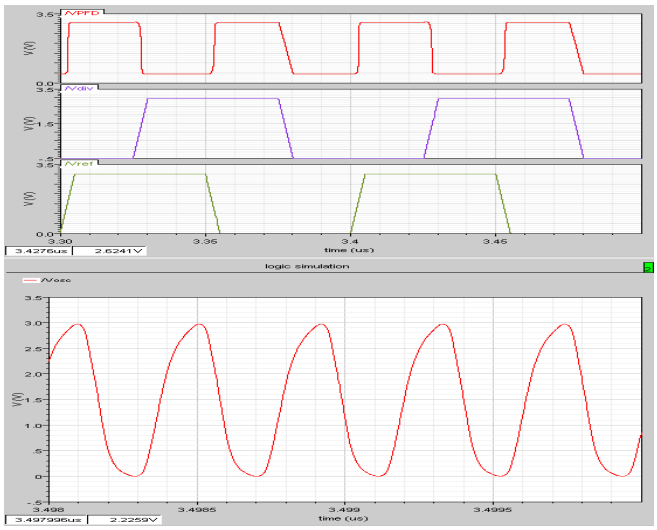

Figure 7. TB "PLL_PB". Details of PLL initial transient response: Voltages VPFD, Vdiv and Vref (top), VCO output voltage Vosc (bottom); time intervals: [3.3µs,3.5µs] and [3.498µs,3.5µs] respectively.

Fig. 8 and Fig. 9 show the results of the Spectre initial transient simulation of the BB TB. Initial conditions and time interval are the same as before. CPU time is 11.4 s, i.e., speedup is about 370. This value is reasonable, because

- The maximal time step in the BB TB is at least N_{div} = 244 times greater than in the PB,
- The number of nodes, equations and bsim3v3 models in the BB TB is smaller than in the PB (see Table I).

The two instantaneous frequencies f_{div} and f_{ref} are also calculated using *freq()*. As expected, f_{ref} is constant (10.0 MHz) and f_{div} approaches f_{ref} at the end of the settling time. Transient response is also oscillating and decays in 3.5 µs. VCO control voltage Vctrl, voltages VPFD, Vdiv, Vref and frequency f_{div} in the locked state are equal in both test benches (see the corresponding graphs on top of Fig. 7 and Fig. 9).

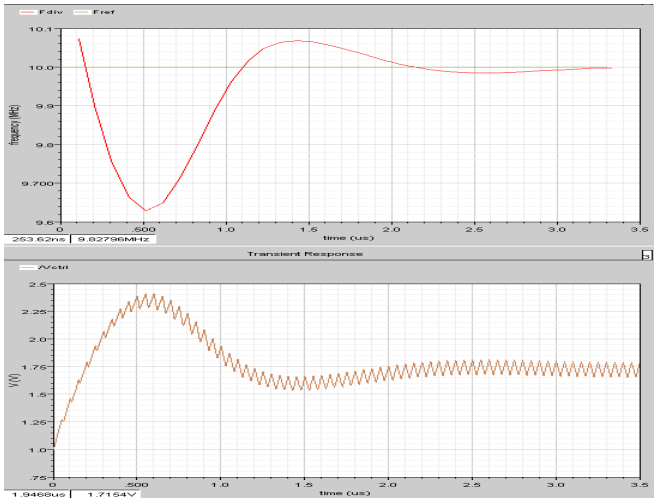

Figure 8. TB "PLL_BB". PLL initial transient response: Frequencies f_{div} and f_{ref} (top), VCO control voltage Vctrl (bottom); time interval: [0, 3.5µs].

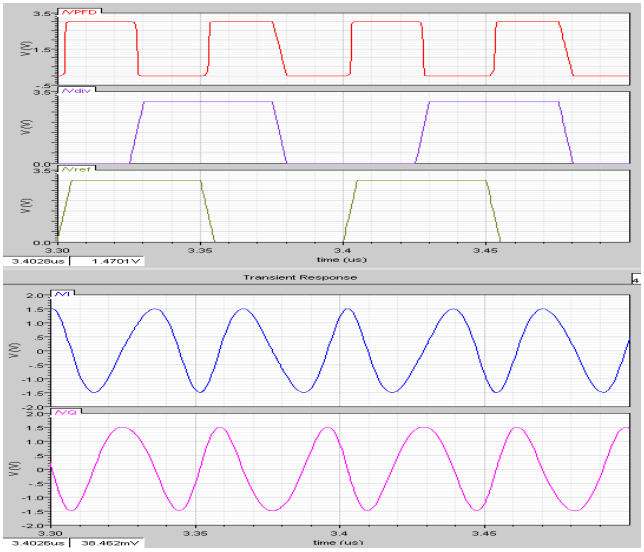

Figure 9. TB "PLL_BB" – details of PLL initial transient response: Voltages VPFD, Vdiv and Vref (top), BB VCO output voltages VI and VQ (bottom); time interval: [3.3µs,3.5µs].

In comparison to the PB TB (Fig. 6), the transient response of the BB TB (Fig. 8) has a small difference at the beginning of simulation – i.e., in time interval [0, 1.1µs]. There are at least two reasons for this difference. Firstly, the BB model of the VCO is based on the PSS simulation results of the VCO transistor level model and therefore does not model its initial transient response. Secondly, the Verilog-A operator in the BB model of the VCO `$table_model(V(ctrl), "VCO.tbl", "1CC")` implements the frequency characteristic $f_{vco}=F(v_{ctrl})$ as a PWL function of v_{ctrl} . The values of f_{vco} are linearly interpolated in the interval [1.125 V, 2.875 V] and clamped if v_{ctrl} < 1.125 V or v_{ctrl} > 2.875 V. Thus, at the beginning of simulation – i.e., in the interval 1.0 V< v_{ctrl} < 1.125 V, f_{vco} in the BB model is constant, causing an additional phase difference between the FD output pulses in both test benches.

TABLE I. COMPARISON OF PB AND BB TEST BENCHES

Parameter	*PB*	*BB*
Nodes	14	13
Equations	58	47
Bsim3v3 models	19	10
PB Verilog-A model of FD	1	-
BB Verilog-A model of FD	-	1
BB Verilog-A model of VCO	-	1
CPU time (s)	4270.0	11.4

The comparison of the results from the PB and BB simulations shows that the calculation of the instantaneous frequency and phase in the BB models of VCO and FD are correct. Baseband modeling of components accelerate the simulation about 370 times with high accuracy.

IV. EXAMPLE: SIMULATION OF FREQUENCY HOPPING IN PLL SYNTHESIZER

PB and BB models of FDs allow the transient simulation of a frequency hopping operation mode of PLL synthesizers changing the FD control voltage VNdiv. We have simulated "worst case frequency jumps" of N_{div} using two different PWL functions of time for the voltage VNdiv.

In the 1st case, the FD control voltage VNdiv invokes two jumps of N_{div}: from 240 to 248 and back at t_{up}=10.0 µs and t_{down}=16.0 µs respectively. Simulation results of the PB and BB test benches in the time interval [0, 25µs] are shown on Fig.10 and Fig.11 respectively. The numbers of steps during transient simulation are $2.46\cdot10^7$ (PB) and $7.10\cdot10^4$ (BB). The VCO control voltages Vctrl of both TBs are equal. Thus, this PLL synthesizer performs well in the 1st "worst case", and finds each locked state after the decay of the corresponding transients.

In the 2nd case, the PWL function invokes two larger jumps of N_{div}: from 240 to 300 and back at the same time points. Simulation results of the PB and the BB test benches in time interval [0, 25µs] are shown in Fig.12 and Fig.13 respectively. The numbers of steps during transient simulation is $2.46\cdot10^7$ and $7.28\cdot10^4$ respectively, also mixed mode simulation with BB models requires also about 340 times less of time steps.

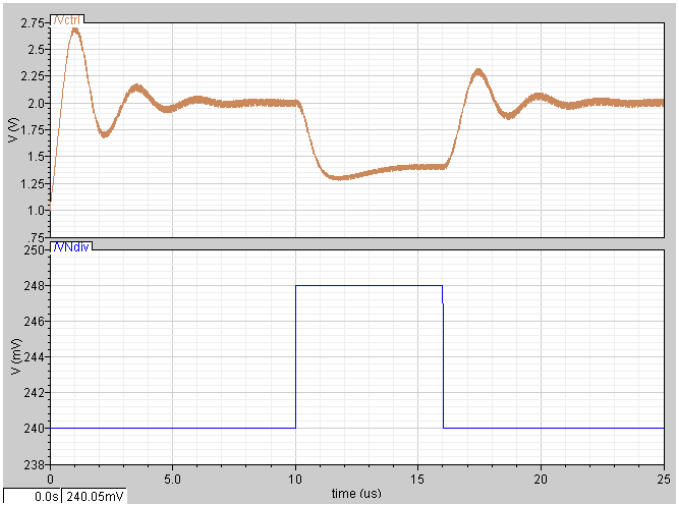

Figure 10. TB "PLL_PB". PLL transient response (1st case of VNdiv): VCO control voltage Vctrl (top), FD control voltage VNdiv (bottom); time interval: [0, 25µs] .

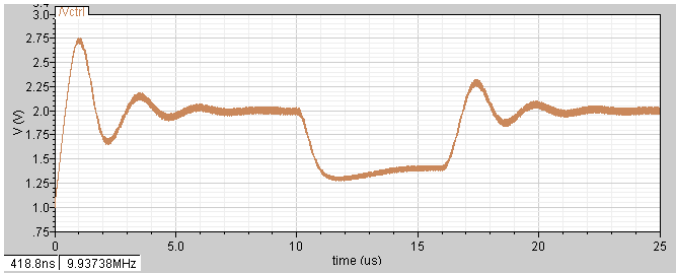

Figure 11. TB "PLL_BB". PLL transient response (1st case of VNdiv): VCO control voltage Vctrl; time interval: [0, 25µs] .

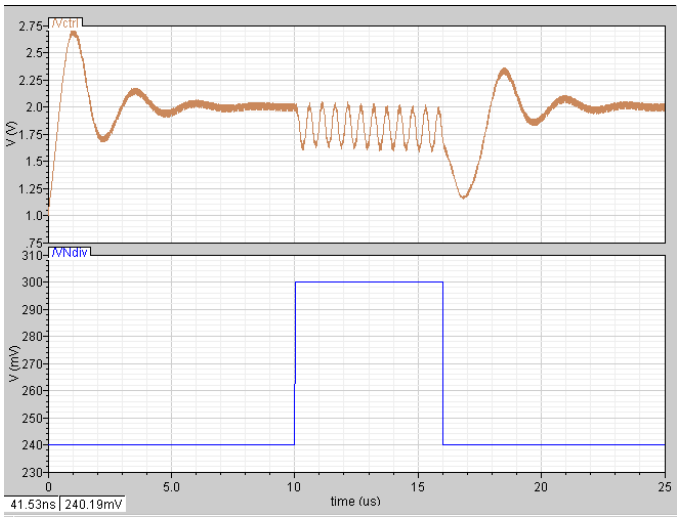

Figure 12. TB "PLL_PB" – PLL transient response (2nd case of VNdiv): VCO control voltage Vctrl (top) , FD control voltage VNdiv (bottom); time interval: [0, 25µs] .

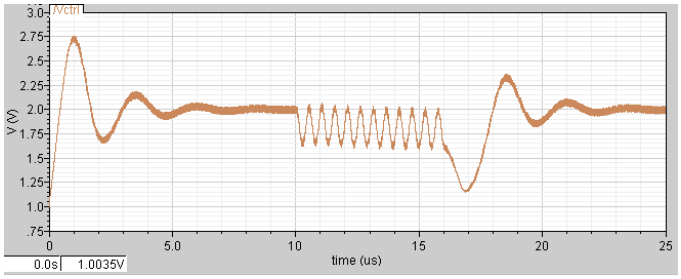

Figure 13. TB "PLL_BB". PLL transient response (2nd case of VNdiv): VCO control voltage Vctrl; time interval: [0, 25µs]

In the 2nd case (Fig. 12 and 13) the PLL synthesizer also performs as expected. VCO control voltage Vctrl used in both test benches are equal. Only in the unlocked during the time interval [10 µs, 16 µs] we have N_{div} = 300, because the maximal frequency of this VCO (2.49 GHz) is less than the frequency, which is set by the FD value $N_{div} \cdot f_{ref}$ = 300·10.0 MHz = 3.0 GHz.

The example shows that the proposed BB models of the VCO and the FD exhibits the frequency hopping behaviour of the PLL at the same accuracy as the PB model reference.

V. CONCLUSIONS

We have presented a new approach to fast mixed-mode PLL simulation based on Spice-like simulators and BB models of VCO and FD. Other PLL blocks (phase-frequency detectors, charge pumps, and loop filters) may be transistor level and/or behavioral PB models. The created BB models of the VCO and the FD can be used as Verilog-A templates. The efficiency and accuracy of the proposed approach is tested on CMOS N-integer frequency synthesizer. The proposed approach allows very fast PLL simulation and optimization of sophisticated PFD and CP blocks on transistor level with speedups of about 2-3 orders of magnitude.

REFERENCES

[1] B. Razavi, "Design of monolithic phase-locked loops and clock recovery circuits - A tutorial, " in "Monolithic phase-locked loops and clock recovery circuits: theory and design", IEEE Press, 1996.

[2] Verilog-AMS Language Reference Manual, Analog & Mixed-Signal Extensions to Verilog HDL, Version 2.3.1, June 1, 2009, Accellera, www.vhdl.org/verilog-ams/htmlpages/public-docs/lrm/2.3.1/VAMS-LRM-2-3-1.pdf .

[3] "Behavioral Modeling of PLL Using Verilog-A," Silvaco, 2003, www.silvaco.com/tech_lib/simulationstandard/2003/jul/a2/a2.html .

[4] Verilog-A Model Library, www.designers-guide.org/VerilogAMS/ .

[5] R. Frevert, et al., Modeling and Simulation for RF System Design, 1st ed., Dordrecht, The Netherlands : Springer, 2005.

[6] P. Van Halen and G.R. Boyle, "SPICE-compatible behavioral phase-space simulation techniques for phase-locked systems," Proceedings of the 38th Midwest Symposium on Circuits and Systems, vol.1, pp.53 - 56, 13-16 Aug. 1995.

[7] X. Lai, Y. Wan, and J. Roychowdhury, "Fast PLL simulation Using Nonlinear VCO Macromodels for Accurate Prediction of Jitter and Cycle-Slipping due to Loop Non-idealities and Supply Noise," Proc. IEEE Asia South-Pacific Design Automation Conference, January 2005.

[8] X. Lai , J. Roychowdhury, "TP-PPV: piecewise nonlinear, time-shifted oscillator macromodel extraction for fast, accurate PLL simulation", Proc. 2006 IEEE/ACM International conference on computer-aided design, November 05-09, 2006, San Jose, California.

[9] M.H. Perrott, "Fast and accurate behavioral simulation of fractional-N synthesizers and other PLL/DLL circuits," Proc. of Design Automation Conference, pp. 498-503, June 10-14, 2002, New Orleans, Louisiana.

[10] S. Sancho, A. Suarez, and J. Chuan, "General envelope-transient formulation of phase-locked loops using three time scales," IEEE Transactions on Microwave Theory and Techniques , vol. 52, no. 4, pp. 1310-1320, 2004.

[11] R. A. Rutenbar, G. G. E. Gielen, and J. Roychowdhury, "Hierarchical modeling, optimization, and synthesis for system-level analog and RF designs," Proceedings of the IEEE, vol. 95, pp. 640-669, March 2007.

[12] M.C. Jeruchim, P. Balaban, and K.S. Shanmugan, Simulation of communication systems, 2nd ed., Kluwer Academic, 2000.

[13] Jesse E. Chen, Modeling RF Systems, 2005, www.designers-guide.org/Modeling/modeling-rf-systems.pdf .

[14] R. Frevert, I. Harasymiv, W. Hartong, J. Hartung, U. Knöchel, "Automatische Generierung templatebasierter HF-Basisband-Modelle," GMM Fachbericht, Analog 2005, S.205-210, Hannover ("Automatic generation of template-based HF baseband models," GMM technical report, Analog 2005, pp.205-210, Hannover), Germany.