Verilog-AMS-PAM: Verilog-AMS integrated with Parasitic-Aware Metamodels for Ultra-Fast and Layout-Accurate Mixed-Signal Design Exploration

Geng Zheng[1], Saraju P. Mohanty[2], Elias Kougianos[3], Oleg Garitselov[4]
NanoSystem Design Laboratory (NSDL)[1,2,3,4]
Dept.of Computer Science and Engineering[1,2,4] and Dept. of Electrical Engineering Technology[3]
University of North Texas Denton, TX 76203.[1,2,3,4]
gengzheng@my.unt.edu[1], saraju.mohanty@unt.edu[2], eliask@unt.edu[3]

ABSTRACT

Current Verilog-AMS system level modeling does not capture the physical design (layout) information of the target design as it is meant to be fast behavioral simulation only. Thus, the results of behavioral simulation can be very inaccurate. In this paper a **paradigm shift of the current trend** is presented that integrates layout level information (with full parasitics) in Verilog-AMS through polynomial metamodels such that system-level simulation of a mixed-signal circuit/system is realistic and as accurate as the true parasitic netlist simulation. As a specific case study, a voltage-controlled oscillator (VCO) Verilog-AMS behavioral model and design flow are proposed to assist fast PLL design exploration. Based on a quadratic polynomial metamodel, the PLL simulation achieves approximately a $10\times$ speedup compared to the layout extracted, parasitic netlist. The simulations using this behavioral model attain high accuracy. The observed error for the simulated lock time and average power consumption are 0.7 % and 3 %, respectively. This behavioral metamodel approach bridges the gap between layout accurate but fast simulation and design space exploration.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles—VLSI (very large scale integration)

Keywords

Polynomial and Nonpolynomial Metamodel, Mixed-Signal Design, Behavioral Simulation, Verilog-AMS Modeling, PLL.

1. INTRODUCTION

Parasitics greatly degrade the performance of nano-CMOS circuit designs. They cause significant mismatch between schematic and layout circuit simulations. To account for the parasitic effects and achieve design closure, numerous iterations at the layout stage are usually required. This process requires great amounts of time and effort. Layout-accurate verification is the major obstacle because the iteration time is mainly spent on layout modification and simulations. Behavioral models that are capable of representing circuit layout have the potential to dramatically shorten the design cycle [8, 7, 12]. Parasitic effects, however, are not discussed in most works due to the inherent inability of function-based behavioral models to account for them. Also, circuit models in these works are commonly implemented as Verilog-A modules rather than Verilog-AMS modules which are more flexible in terms of functionality. Modeling techniques such as model order reduction [14] and symbolic model generation [3] have been also proposed but they only work for relatively small circuits. It may be noted that **the terms macromodel and metamodel are often used interchangeably** in the literature. However, while macromodels are simplified models of a circuit and system that use the same simulator [2], metamodels are mathematical algorithms that can decouple the design and simulations to a pure behavioral tool such as MATLAB [4].

A metamodeling technique for nano-CMOS AMS circuits was proposed in [5]. The models built with this method accurately reflect parasitic effects. In the present work, an accurate VCO behavioral model is proposed based on this approach. This behavioral model is implemented using the Verilog-AMS language which enables fast simulations. Combining the metamodeling technique and Verilog-AMS simulation, the design verification process achieves a large speedup and maintains reasonably high accuracy. In fact, not only the proposed Verilog-AMS behavioral model can help the design space exploration and optimization, it can also assist the verification process of complex System-on-Chip (SoC) designs. A phase-locked loop (PLL) design with an LC-tank VCO using 180 nm CMOS process is used to demonstrate the modeling technique, design flow and implementation method. Among different PLL architectures, the charge-pump PLL (CPPLL) has been widely used in various system due to its simplicity and effectiveness.

The **novel contributions** of this paper are as follows:

1. An accurate and efficient quadratic polynomial metamodel for a 180 nm LC VCO design is developed.

2. A Verilog-AMS module is constructed to implement the VCO metamodel.

3. A parameterized netlist approach using the VCO layout netlist after full parasitic extraction is used to capture parasitic effects by the metamodel.

4. Metamodel-integrated PLL simulations are presented and the accuracy and speed of the proposed VCO behavioral Verilog-AMS model are discussed.

5. A metamodel-assisted PLL optimization flow is demonstrated.

GLSVLSIŠ12, May 3–4, 2012, Salt Lake City, Utah, USA.
Copyright 2012 ACM 978-1-4503-1244-8/12/05 ...$10.00.

The rest of this paper is organized as follows: Section 2 discusses previous works relevant to PLL behavioral modeling. Section 3 describes the metamodeling technique and the proposed Verilog-AMS VCO behavioral module. Section 4 presents the PLL simulation flow and methodology with the proposed VCO behavioral model. Section 5 demonstrates the PLL optimization with the assistance of the metamodel. Section 6 concludes this work and discusses future research.

2. RELATED PRIOR RESEARCH

Verilog-A behavioral modules of linear VCOs were used in [9] for PLL jitter characterization and in [15] for aiding a hierarchical CPPLL sizing method. No parasitic effects were included in this model. A characterization technique is developed in [11] to extract circuit parameters, including parasitic effects. The authors also adopted the linear VCO model which may be sufficient for performing verification on fixed designs, but is not useful for design exploration since the VCO linearity condition is not always valid. The VCO behavioral models developed in [1] and [6] use a table-lookup approach inside Verilog-A modules, which is not efficient for global design space exploration. An event-driven analog modeling approach was proposed in [13] which used the Verilog-AMS `wreal` data type to improve the model efficiency. However, it is not clear how the VCO gain and output frequency were modeled.

3. LAYOUT-ACCURATE POLYNOMIAL METAMODELING OF CPPLL

3.1 High-level Description of the CPPLL

A typical CPPLL consists of a phase/frequency detector (PFD), a charge-pump (CP), a loop filter (LF), and a VCO. If the PLL needs to perform frequency synthesis, a frequency divider (FD) will also be employed. The system level topology of a CPPLL is shown in Fig. 1. In this paper, we focus on developing a VCO behavioral model that can accurately mimic the VCO *physical* design. The model is constructed using the Verilog-AMS language to enable fast design exploration. The other parts of the PLL are modeled with hardware description languages or at schematic level in order to simulate the whole PLL system.

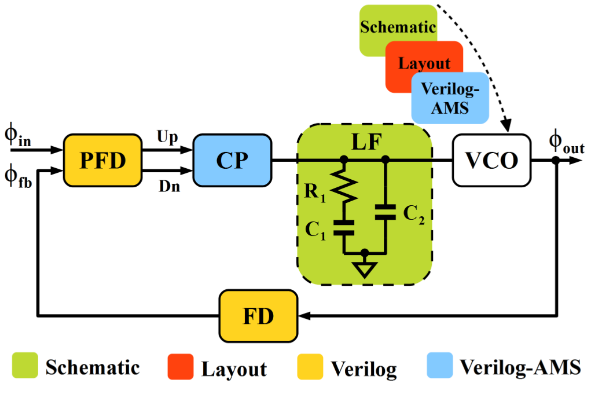


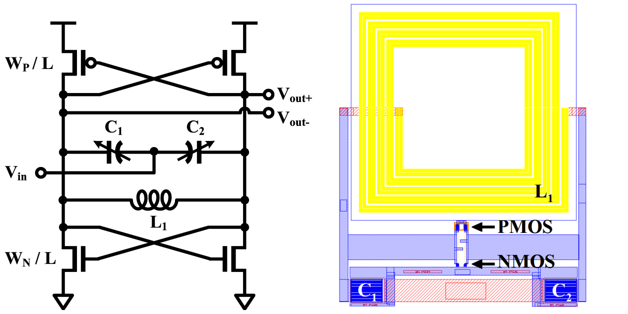
Figure 1: The CPPLL configuration in this paper.

3.2 CPPLL Verilog-AMS Behavioral Model

Mixed-signal systems such as CPPLLs can be simulated using mixed-signal simulators which have two kernels—an event-driven digital kernel and a continuous-time analog kernel [10]. Calling the analog kernel in a mixed-signal simulation is generally far more computationally expensive than calling the digital kernel. Thus for fast design verification with given accuracy requirements, models that will cause unnecessary analog events should be avoided.

Fig. 1 illustrates the CPPLL configuration in this work. The LF consists of three simple passive components R_1, C_1, and C_2. Modeling it behaviorally does not improve the simulation efficiency noticeably. Therefore the SPICE model is used for the LF and it is implemented in a schematic view. The PFD and FD are pure digital circuits. The frequency of the FD output ϕ_{fb} is $1/N$ of that of the VCO output ϕ_{out}, where N is the FD division ratio. The PFD activates its output Up or Dn to vary the VCO output until ϕ_{fb} and ϕ_{in} are aligned and have the same frequency. They introduce non-idealities to the system via their signal delay, and the rise/fall time. These non-idealities can be easily described in the digital domain. Thus the behavior of these two blocks is implemented using the Verilog language. The CP has digital inputs and analog output so it is implemented as a Verilog-AMS module.

Three different views have been implemented for the VCO: (1) schematic, (2) layout with parasitics, and (3) Verilog-AMS. Fig. 2 shows the schematic and layout views of the LC VCO design. Both schematic and layout views use SPICE models for simulation. While the layout view includes the parasitic elements therefore takes longer to simulate, it results in accurate estimate of the real silicon. Table 1 lists the number of elements in the schematic view and parasitic extracted layout view. The parasitics consist of Resistance (R), Capacitance (C), and self (L) and mutual inductance (K).



((a)) Schematic view ((b)) Layout view

Figure 2: The LC VCO schematic and layout views. $L = 180$ nm; $W_P = 20$ μm; $W_N = 10$ μm.

Table 1: Element Counts for LC VCO Schematic and Layout.

Elements	**Schematic**	**Layout**
Transistor	4	4
Inductor	1	10
Capacitor	2	38
Resistor	0	560
Total	7	612

The Verilog-AMS view implements an accurate behavioral model. The modeling approach is detailed in Section 3.3.

3.3 VCO Polynomial Metamodeling

The VCO behavior is mainly determined by its voltage frequency transfer curve. A common way to model a VCO to assume that the

VCO is perfectly linear and model it with the following:

$$f_{osc} = f_0 + K_{VCO} V_C, \tag{1}$$

where f_{osc} is the oscillation frequency, f_0 is the center frequency, K_{VCO} is the gain, and V_C is the control voltage at the VCO input. This linear model can be implemented by sampling two data points on the VCO transfer curve. When performing design exploration, however, the linearity is not guaranteed, which leads to invalid simulation results. Also, parasitic effects from layout extraction further degrade the accuracy of this modeling approach. To account for the non-linearity and layout parasitics, the metamodeling approach suggested in [4] is used.

We chose to implement polynomial metamodels because they have the following advantages: (1) they are simple closed form equations which are easy to implement; (2) their form is flexible so that one can quickly examine and compare the accuracy of polynomial models with different degree; (3) they have been widely used and their properties are well understood. The polynomial metamodel used in this paper is as follows:

$$f(\mathbf{x}) = \sum_{i=0}^{K-1} \beta_i {x_1}^{p_{1i}} {x_2}^{p_{2i}} {x_3}^{p_{3i}}, \tag{2}$$

where x_1, x_2, and x_3 are three input variables corresponding to W_P, W_N, and V_C in this work, respectively. K is the number of basis functions this model has and β_i is the coefficient for the basis function. $f(\mathbf{x})$ is the output that approximates the true model. In order to construct the metamodel for a given VCO design, for each basis function the coefficient β_i and the power terms p_{1i}, p_{2i}, and p_{3i} for each input variable need to be obtained. This is done in three steps: first, a set of input variables $[x_1\ x_2\ x_3]$ is generated using the Latin Hypercube Sampling (LHS) technique; second, circuit simulations are performed and the outputs for each set of inputs are saved; third, with the inputs and outputs from previous steps, the coefficients and the power terms that lead to a model with good fit are computed. In order to incorporate the parasitic effects into the model without repeating the layout for each simulation, the netlist for the extracted layout view is parameterized for W_P and W_N.

In this work, we are interested in the VCO output frequency and its power consumption. Therefore two respective metamodels are built. They share the same power terms for the input variables, while the coefficients β_i in the two models are different. After these values are computed, they are written into a text file which will be read by the VCO Verilog-AMS module to implement the model. A quadratic polynomial metamodel with first order interaction has been implemented. Table 2 shows the layout of the text file storing the values for the power terms and the coefficients for this model obtained from 100 samples. In the table, $\beta_{i,f}$ and $\beta_{i,p}$ are the coefficients for the frequency and power consumption models, respectively. These values are read into the Verilog-AMS module during the initialization process.

Fig. 3 shows a portion of the VCO Verilog-AMS module. The part of the basis function related to the input variables W_P and W_N is constructed in the `initial` block. The remainder of the basis functions are constructed in the `always` block since the third variable V_C needs to be updated continuously during the simulation. The output signal of this module is implemented to be digital logic type to reduce the computation cost. As in the PFD and FD modules, the non-idealities associated with this output signal can be modeled in the digital domain.

This Verilog-AMS module can be easily reconfigured for metamodels with different degrees by changing the parameter K. In Fig. 4, the simulation results of the VCO transfer curves for the design in Fig. 2 are shown. The parasitics cause a large difference between the schematic and layout results both in the VCO center frequency

Table 2: Layout of the text file storing the power terms and coefficients for the VCO quadratic polynomial metamodel

i	p_{1i}	p_{2i}	p_{3i}	$\beta_{i,f}$	$\beta_{i,p}$
0	0,	0,	0,	2.113e+009,	1.385e-005
1	1,	0,	0,	-3.214e+012,	44.459e+000
2	2,	0,	0,	3.456e+016,	-2.804e+005
3	0,	1,	0,	6.869e+012,	39.729e+000
4	1,	1,	0,	-1.021e+017,	2.911e+005
5	0,	2,	0,	-2.071e+017,	-1.080e+006
6	0,	0,	1,	3.513e+008,	-8.271e-004
7	1,	0,	1,	-2.565e+012,	-31.282e+000
8	0,	1,	1,	-5.331e+012,	-11.392e+000
9	0,	0,	2,	0.000e+000,	1.041e-003

```
‘timescale 10ps / 1ps
‘include "disciplines.vams"
module vco_metamodel (out, in);
... ...
parameter integer K;
initial
begin
    out = 0;   // Initialize vco digital output
    ... ... // Declare ports and data types
    metaf = $fopen("metamodel.csv", "r");
    while (!$feof(metaf))
    begin
        readfile = $fscanf(metaf,
                   "%e,%e,%e,%e,%e\n",
                   p1, p2, p3, betaf, betap);
        bf[i] = pow(wp,p1) * pow(wn,p2) * betaf;
        bp[i] = pow(wp,p1) * pow(wn,p2) * betap;
        pv[i] = p3;
        i = i + 1;
    end
    $fclose(metaf);
    ... ...
end
always
begin
    vc = V(in);
    ... ...
    freq = 0;
    power = 0;
    for (i = 1; i <= K; i = i + 1)
    begin
        freq = freq + bf[i] * pow(vc, pv[i]);
        power = power + bp[i] * pow(vc, pv[i]);
    end
    ... ...
    #(0.5 / freq / 10p)
    out = ~out;
end
... ...
endmodule
```

Figure 3: Example of the VCO Verilog-AMS source code implementing the polynomial metamodel.

and the gain. Metamodel 1 is the Verilog-AMS module with the quadratic model from 100 samples. Metamodel 2 is the module with a 5-th degree polynomial model from 500 samples. Metamodel 2 does not have significant improvements over Metamodel 1. Thus Metamodel 1 is used in the PLL simulations shown in Sections 4 and 5. Differences between the transfer curves of layout and metamodel Verilog-AMS views can still be observed, which means a better metamodel may be used to further improve the accuracy. However, as will be seen in Section 4, this polynomial metamodel is sufficient for system level PLL verification to simulate lock time and average power dissipation.

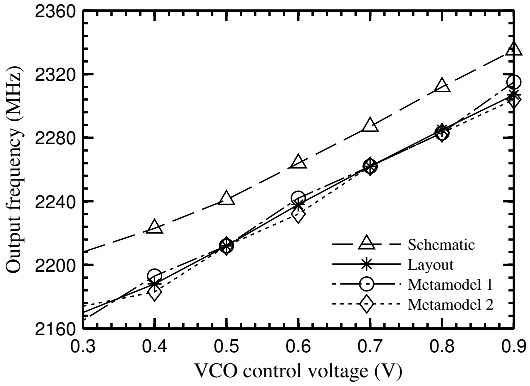


Figure 4: VCO transfer curves for threes different views.

4. VERILOG-AMS-PAM BASED SIMULATION OF PLL

In this section, we demonstrate PLL simulations with the VCO design shown in Fig. 2. The PLL configuration shown in Fig. 1 is used. The PFD and FD are in Verilog view. The CP is in Verilog-AMS view and the LF is in schematic view. The views for the aforementioned blocks were not changed throughout all simulation runs. The VCO view was changed from schematic, to layout with parasitics, and then to Verilog-AMS views. Two Verilog-AMS views have been implemented—one for the linear model and one for the quadratic metamodel proposed in Section 3.3. The results for using different VCO views are compared.

A 550 MHz input clock ϕ_{in} is assigned to the PLL input. The FD has a division ration of 4. Thus the desired frequency for the PLL output clock ϕ_{out} is 2200 MHz. Fig. 5 shows the ϕ_{out} frequencies from 500 ns transient simulations with different VCO views. Although the PLLs with the different VCO views are all able to lock to the same correct frequency, the one with the schematic view shows quite different locking behavior compared to the one with the layout views. This mismatch is due to the parasitic effects which greatly change the VCO characteristics. The one with the linear model shows improvements over the the schematic since the parasitics have been taken into account. However, it still has significant errors, for example, in the lock time. The PLL with the metamodel Verilog-AMS view offers the best approximation of the true model and accurately estimated the lock time. To further understand the behavior of the PLL with different VCO views, the critical analog signal V_C was inspected.

Fig. 6 compares the V_C waveform from the four simulations. Again, the metamodel Verilog-AMS view provides the best approximation of the layout view behavior. The PLL with the schematic VCO view can just barely lock to 2200 MHz since V_C is approaching the NMOS threshold. This shows that the center frequency and the gain of the schematic VCO are very different from the layout. These also confirm the VCO transfer curves plotted in Fig. 4.

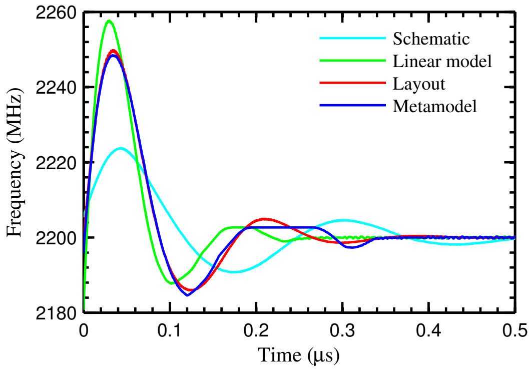


Figure 5: PLL output frequency from AMS simulation with three different VCO views.

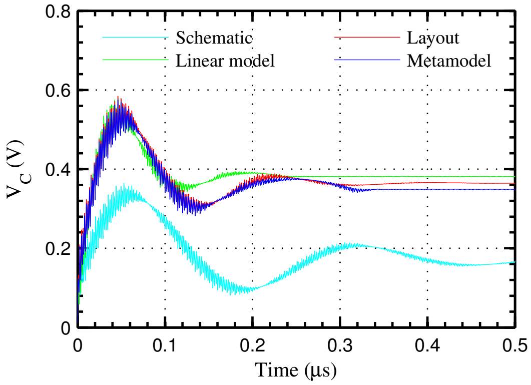


Figure 6: VCO control voltage waveforms from PLL simulations.

The Verilog-AMS metamodel also facilitates estimation of the power consumption. Fig. 7 shows the average VCO power consumption per fifty cycles in the four simulations. It once again confirms that the Verilog-AMS metamodel can better model the layout counterpart. Table 3 shows the PLL simulation results to compare the accuracy of the linear model and the proposed metamodel.

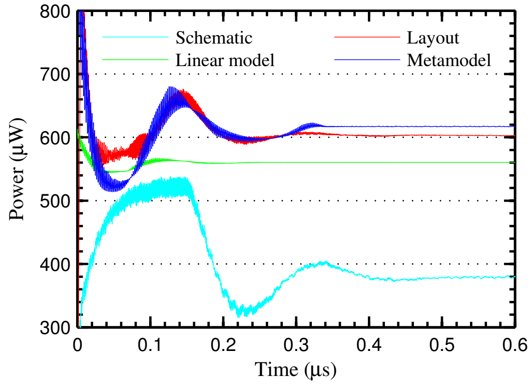


Figure 7: Average VCO power consumption per 50 cycles

In Table 3, the estimated PLL lock time is listed. The one from the simulation with the VCO layout view serves as the true model. The errors resulting from the other two models are computed. The metamodel achieves a very low error rate of $0.7\ \%$, while the linear model causes a large error of $31.7\ \%$. $\mathbf{f_{Locked}}$ is the PLL output frequency when it is locked. $\mathbf{P_{Locked}}$ is the average VCO power consumption when the PLL is locked. Again, the metamodel resulted in a good estimate of the true power dissipation. The $\mathbf{V_C}$ root-mean-square error (RMSE) of the models for the 500 ns simulations are also listed.

Table 3: Comparison of PLL Simulations with Different VCO Modules

	Layout	**Linear Model**	**Metamodel**
Lock time (ns)	335.4	229.1	332.9
Error %	0.0 %	31.7 %	0.7 %
$\mathbf{f_{Locked}}$ (MHz)	2199.99	2199.99	2199.99
Error %	0.0 %	0.0 %	0.0 %
$\mathbf{P_{Locked}}$ (μW)	602	560	620
Error %	0.0 %	7.0 %	3.0 %
$\mathbf{V_C}$ RMSE (mV)	0	33.508	10.889

Table 4 compares the runtimes for the PLL transient simulations with the layout, schematic, and Verilog-AMS metamodel VCO views. The Verilog-AMS metamodel achieves roughly a $10\times$ speedup compared to the layout. Note that in practice the VCO design may contain more complex circuitry which leads to longer runtime for a simulation run. The runtime for simulation with the Verilog-AMS module will not be different. Thus the speedup will be more significant in that case.

Table 4: Comparison of The Speed of The PLL Simulations with Different VCO Modules

	Layout	**Schematic**	**Metamodel**
Runtime	80.5 s	40.3 s	8.7 s
Normalized speed	$1\times$	$\sim 2\times$	$\sim 10\times$

5. PLL OPTIMIZATION USING METAMODELS

The metamodel and its Verilog-AMS implementation can also accelerate design optimization. In this section, we demonstrate how the metamodel assists PLL optimization. Modern low-power devices are been used in many applications. The wake up time for these device is crucial, which requires short lock time if PLLs are employed. The goal of this optimization is to find a design with minimized lock time and low power consumption. The transistor sizes W_P and W_N of the LC VCO are chosen as the design variables to be optimized. A simple optimization flow is developed to highlight the use of the metamodel and its Verilog-AMS implementation. In practice, more sophisticated flows can be used to handle problems of larger sizes. Table 5 summarizes the optimization flow.

In the first step, the ranges of the design variables W_P and W_N are defined to be 10–30 μm and 5–15 μm, respectively. Within each range, 31 values are evenly selected, which results in a total of 961 possible designs. The design space is then reduced by applying the tuning range constraint. We define the desired VCO frequency tuning range to be 2180–2300 MHz. A metamodel is used in this step to calculate the VCO tuning range for each design without performing circuit simulations. Only 320 designs are left after this step. Verilog-AMS simulations are then run to obtain the PLL lock time for these designs. The simulations only took 30 minutes to complete due to the use of the Verilog-AMS module. The top five designs with the minimum lock time are saved. These designs are listed in Table 6 along with their average power consumption when the PLL is locked.

Table 5: Summary of the optimization flow

Step #	**Action**		**Design Space** (total design counts)
1	Define design space	→	961
2	Shrink design space with tuning range constraint	→	320
3	Run AMS simulation to obtain design choices with minimized lock time	→	5
4	Select optimal design with low-power consideration	→	1
5	Verify the final design with layout simulation	→	Done

Table 6: Comparison of the choices for optimal design

Choice #	W_P (μm)	W_N (μm)	**Lock Time** (ns)	$\mathbf{P_{Locked}}$ (μW)
1	23.2	5	328.6	504
2	21.4	5	328.7	486
3	21.4	5.3	330.4	494
4	22	5	330.4	492
5	22.6	5.3	330.4	506

Choice 2 from Table 6 is selected as the final design for its lowest power consumption. Fig. 8 shows the top five design candidates in the design space of 961 designs. Although the lock time can be further minimized the resulting designs would violate the tuning range requirements. Table 7 compares the original LC VCO design (baseline) shown in Fig. 2 and the optimal design. The optimization reduces both the lock time and the power consumption. Fig. 9 shows the PLL simulation with the VCO layout view of the optimal design relocks from 2180 MHz to 2300 MHz. This simulation finalizes the verification of the optimal design.

6. CONCLUSIONS

A Verilog-AMS behavioral model based on quadratic polynomial metamodeling for a 180 nm LC VCO has been proposed. With this behavioral model, fast and accurate PLL design verification and optimization have been demonstrated. The behavioral model can be further improved but is sufficient for lock time and average power estimation. Future research includes developing behavioral

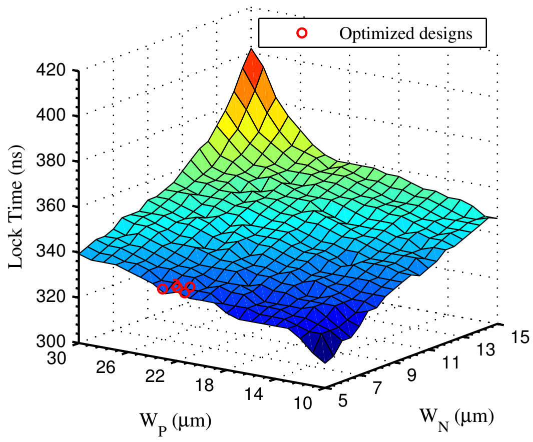


Figure 8: Sizing the transistors for lock time.

Table 7: Comparison of baseline and optimized designs

	Baseline	Optimal	Reduction
W_P/W_N (μm/μm)	20 / 10	21.4 / 5	–
Lock time (ns)	335.4	320.4	15.0
$\mathbf{P_{Locked}}$ (μW)	602	455	147
Tuning Range (MHz)	2170–2304	2173–2321	–

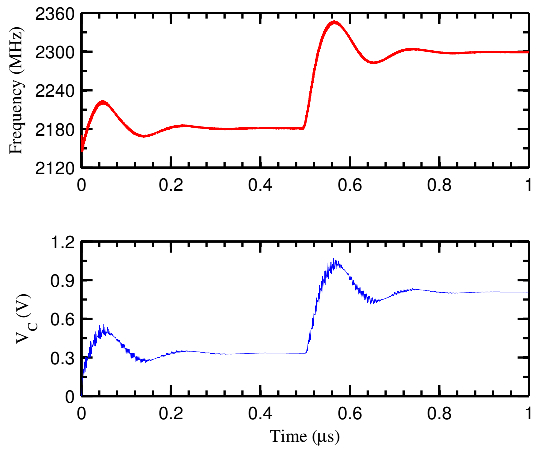


Figure 9: Simulation showing that the PLL first locks to 2180 MHz and then relocks to 2300 MHz.

models that incorporate parasitics for the rest of the PLL building blocks and studying different metamodeling methods.

7. REFERENCES

[1] S. Ali, L. Ke, R. Wilcock, and P. Wilson. Improved performance and variation modelling for hierarchical-based optimisation of analogue integrated circuits. In *Proc. DATE '09. Design, Automation & Test in Europe Conf. & Exhibition*, pages 712–717, 2009.

[2] S. Basu, B. Kommineni, and R. Vemuri. Variation-Aware Macromodeling and Synthesis of Analog Circuits Using Spline Center and Range Method and Dynamically Reduced Design Space. In *Proc. Int. Conf. VLSI Des., VLSID*, pages 433–438, 2009.

[3] C. Borchers. Symbolic behavioral model generation of nonlinear analog circuits. 45(10):1362–1371, 1998.

[4] O. Garitselov, S. P. Mohanty, and E. Kougianos. A Comparative Study of Metamodels for Fast and Accurate Simulation of Nano-CMOS Circuits. *IEEE Transactions on Semiconductor Manufacturing*, 2012.

[5] O. Garitselov, S. P. Mohanty, E. Kougianos, and P. Patra. Nano-cmos mixed-signal circuit metamodeling techniques: A comparative study. In *Proc. Int Electronic System Design (ISED) Symp*, pages 191–196, 2010.

[6] I. Harasymiv, M. Dietrich, and U. Knochel. Fast mixed-mode PLL simulation using behavioral baseband models of voltage-controlled oscillators and frequency dividers. In *Proc. XIth Int Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD) Workshop*, pages 1–6, 2010.

[7] M. Hsieh and G. E. Sobelman. Modeling and verification of high-speed wired links with Verilog-AMS. In *Proc. IEEE Int. Symp. Circuits and Systems ISCAS 2006*, pages 2105–2108, 2006.

[8] C. Hung, W. Wuen, M. Chou, and K. A. Wen. A unified behavior model of low noise amplifier for system-level simulation. In *Proc. 9th European Conf. Wireless Technology*, pages 139–142, 2006.

[9] Q. Jing, T. Riad, and S. Chan. Characterizing PLL jitter from power supply fluctuation using mixed-signal simulations. In *Proc. 2nd Asia Symp. Quality Electronic Design (ASQED)*, pages 112–117, 2010.

[10] K. Kundert and O. Zinke. *The Designer's Guide to Verilog-AMS*. Kluwer Academic Publishers, Boston, 1st edition, May 2004.

[11] C. Kuo. An efficient approach to build accurate behavioral models of PLL designs. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E89-A(2):391–398, Feb. 2006.

[12] S. Pam, A. K. Bhattacharya, and S. Mukhopadhyay. An efficient method for bottom-up extraction of analog behavioral model parameters. In *Proc. 23rd Int. Conf. VLSI Design VLSID '10*, pages 363–368, 2010.

[13] Y. Wang, C. Van-Meersbergen, H.-W. Groh, and S. Heinen. Event driven analog modeling for the verification of PLL frequency synthesizers. In *Proc. IEEE Behavioral Modeling and Simulation Workshop BMAS 2009*, pages 25–30, 2009.

[14] J. Wood, D. E. Root, and N. B. Tufillaro. A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems. 52(9):2274–2284, 2004.

[15] J. Zou, D. Mueller, H. Graeb, U. Schlichtmann, E. Hennig, and R. Sommer. Fast automatic sizing of a charge pump phase-locked loop based on behavioral models. In *Proc. IEEE Int. Behavioral Modeling and Simulation Workshop BMAS 2005*, pages 100–105, 2005.