Particle Swarm Optimization over Non-Polynomial Metamodels for Fast Process Variation Resilient Design of Nano-CMOS PLL

Oleg Garitselov[1], Saraju P. Mohanty[2], Elias Kougianos[3], Geng Zheng[4]
NanoSystem Design Laboratory (NSDL)[1,2,3,4]
Department of Computer Science and Engineering[1,2,4]
Department of Electrical Engineering Technology[3]
University of North Texas, Denton, TX 76203.[1,2,3,4]
omg0006@unt.edu[1], saraju.mohanty@unt.edu[2], eliask@unt.edu[3]

ABSTRACT

An automated top-down design flow to achieve physical design of Analog/Mixed-Signal Systems-on-Chip (AMS-SoCs) is difficult, especially for nano-CMOS. Process variation effects have profound impact on the performance of silicon versus layout design. In this paper metamodels, (surrogate models) and Particle Swarm Optimization (PSO) have been combined in an automated physical design flow for fast design exploration of AMS-SoCs. Neural network based non-polynomial metamodels that handle large numbers of design parameters, are used to predict the statistical process variation effects instead of exhaustive Monte Carlo simulations. The PSO algorithm is used for optimization of the AMS-SoC components using their metamodels instead of the actual circuit. The PSO algorithm followed a two step approach: local and global. The physical design of a Phase Locked Loop (PLL) is considered as a case study circuit. The proposed design flow is approximately 5 times faster while the error is under 2% compared to the Monte Carlo analysis.

Categories and Subject Descriptors

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

Keywords

Neural Network Metamodel, Mixed-Signal Design, PLL

1. INTRODUCTION AND MOTIVATION

Modern consumer electronic devices are Analog/Mixed-Signal Systems-On-Chips (AMS-SoCs). Design of these AMS-SoCs presents specific challenges as the design of analog and digital circuits involves two distinct approaches. For example, the digital design is performance driven, whereas the analog design is specification driven. When AMS-SoCs are fabricated using nano-CMOS, their circuits are strongly impacted by the imperfections of manufacturing processes [1]. However, design decisions are often based on nominal (rather than statistical) values of circuit attributes under the assumption that all transistors are alike. Thus, design decisions based on nominal data are not correct because the used data are either overestimations or underestimations of actual (silicon) data (i.e. design-to-silicon gap).

Process variations have an increasing effect on circuits as the technology is scaling past 100 nm [3],[2] with more dramatic effect on analog circuits [5]. It is essential to produce a process variation resilient design to increase the production throughput and reduce the chip cost.

The simulation analysis for process variation is usually done using a large number of Monte Carlo simulations. To reduce the amount of time it takes to produce the analysis, different techniques have been introduced such as symbolic analysis, hierarchical statistical analysis, and regression based approaches. For a simple circuit, it is possible to produce a process robust design just from the understanding of the behavior of that circuit. For large circuits it is time consuming to run large amount of simulations for process variation analysis due to high complexity of the circuit. The primary goal of this paper is to reduce the time-to-market constraints that are enforced on the design time to create a process resilient design of the circuit. This paper proposes the use of neural network based metamodels that can capture the circuit output in small and large ranges of the design parameters which can then be used for process variation analysis and also for circuit optimization. Then, a Particle Swarm Optimization (PSO) algorithm performs the design exploration over the non-polynomial metamodels to quickly converge to a target design.

2. CONTRIBUTIONS & PRIOR RESEARCH

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

- A novel design flow that combines non-polynomial metamodels and particle swarm optimization for fast nanoscale process variation resilient mixed-signal design exploration.
- Accurate neural network based metamodels are proposed for frequency, power, jitter and locking time of the PLL system.
- Statistical process variation analysis over the metamodels instead of the actual circuit are performed and are shown to be much faster without significant loss of accuracy.
- A particle swarm optimization algorithm is presented for global optimization and process variation analysis.

GLSVLSIŠ12, May 3–4, 2012, Salt Lake City, Utah, USA.

Design optimization approaches to produce process variation tolerant design have being proposed in the existing literature. In [4], variability is estimated for frequency acquisition in digitally controlled oscillators. In [10], PVT-tolerant PLLs are proposed. A PVT-tolerant amplitude controller is proposed in [7] to minimize the phase noise of an LC-VCO. A PVT-tolerant low-jitter digital PLL is presented in [6]. The above circuit-specific approaches are not top-down AMS-SoC design flows and cannot handle large circuits with full-blown parasitics as the exclusive use of SPICE does not make them easily scalable. The current paper will thus significantly advance the state-of-the art in AMS-SoC design exploration. Particle swarm optimization (PSO) has been successfully used on op-amp optimization in [9]. The design of an RF CMOS distributed amplifier is done in [8] using the PSO algorithm.

3. PROPOSED DESIGN FLOW USING NON-POLYNOMIAL METAMODELS AND PARTICLE SWARM OPTIMIZATION (PSO)

3.1 The Proposed Fast/Accurate Design Flow

This section briefly discuses the proposed fast design flow for fast and yet accurate process variation resilient optimization of the mixed-signal systems. After the creation of the physical design, the design characteristics of the PLL has changed from the schematic, as expected. Usually comprehensive manual design iterations follow to adjust the physical design to bring the circuit back to the desired specifications. Then Monte Carlo or corner analysis follows to conduct process variation study. If the circuit does not pass the process variation specifications then more manual labor is needed to adjust the design and more simulations are required for this iterative flow. The proposed design flow uses neural network based metamodels that predict the characteristics of the PLL. The physical design parametric netlist is parameterized and Latin Hypercube Sampling (LHS) is used to create two data sets for training and verification of the neural network model. The neural network based non-polynomial metamodels are created based on the training data set and then verified. The accuracy of the metamodels is required to be very high to be able to predict process variation effects. For each particle in the particle swarm optimization algorithm, a Monte Carlo analysis is run on the model if the output of the model is within the optimization constraints of frequency. This optimization effectively can optimize the circuit on a global scale and also account for process variation of the circuit. The final parameters are then used to adjust the physical design. This design flow reduces the amount of manual labor down to two physical design iterations considerably reducing the design process time.

3.2 The Case Study Circuit PLL in Brief

The PLL, which is shown in Fig. 1, provides a good example of a mixed signal system and is a good candidate for the application of our methodology.

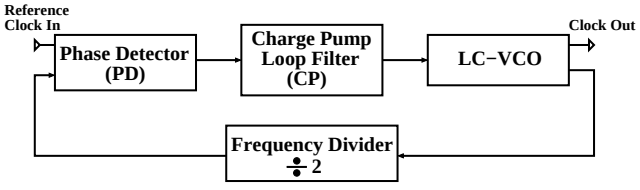


Figure 1: Block diagram of a phase locked loop.

The PLL has distinct components performing different functions and hence their parameters need to be tuned for process variation analysis and optimization. The following design and process parameters were considered: For the LC-VCO, W_{nLCVCO} and W_{pLCVCO}, the width of NMOS and PMOS, L the common length of both transistors, and T_{oxn} and T_{oxp} for the oxide thickness of the NMOS and PMOS. For the divider, Wn_{DIV} and Wp_{DIV}, the width of NMOS and PMOS, L the common length of both transistors, and T_{oxn} and T_{oxp} for the oxide thickness of the NMOS and PMOS. For the charge pump, Wn_{CP} and Wp_{CP} the width of NMOS and PMOS, and L the common length of both transistors. For the phase detector, Wn_{PD} and Wp_{PD} the width of NMOS and PMOS, and L the common length of both transistors.

4. PROCESS VARIATION ANALYSIS USING NEURAL NETWORK METAMODELS

4.1 Non-polynomial metamodeling using feed forward neural networks

Neural network models are composed of a mass of fairly simple computational elements and rich interconnections between them. They operate in a parallel and distributed fashion which resembles biological neural networks. Most neural networks have some form of initial "training" rule in which the weights of connections are adjusted on the basis of presented patterns.

A multiple layer neural network consists of an input, a nonlinear activation function in a hidden layer, and a linear activation function in the output layer. This makes multilayer networks very flexible and powerful due to their ability to represent nonlinear as well as linear functions. The multilayer network needs to have at least one non-linear function layer, otherwise a composition of linear functions becomes just another linear function. The linear layer function output is:

$$v_i = \sum_{i=1}^{s} w_{ji} x_i + w_{j0}, \tag{1}$$

where w_{ji} is the weight connection between the jth component in the hidden layer and the ith component of the input x_i. The nonlinear $\tanh$ activation function used for the hidden layer has the following format: $b_j(v_j) = \tanh(\lambda v_j)$. The network training is performed to minimize the least squares criterion between the predicted ($\hat{y_k}$) and actual (y_k) responses: $E = \sum_{k=1}^{n} (y_k - \hat{y_k})^2$.

The input data set is generated from SPICE simulations, is the same for every metamodel and is generated using LHS. LHS supports any amount of planes and is proven to work better than Monte Carlo due to the more even distribution of points with still the random factor that helps to detect nonlinearity. LHS divides each plane (parameter) into Latin squares and randomly picks a point from each square. Output is generated for each run from a SPICE simulation saving each needed value to its own data set. Hence, each metamodel will have its own target data set. Since the input data set has a large dynamic range, either normalization or standardization of the input data can improve numerical stability. If not, the training of higher values can outweigh the lower and neural network will not train properly. In this paper, the data sets are normalized to mean 0 and standard deviation 1, as experimental studies showed that the normalization is easier to handle than standardization of data even though neural networks performed much better with either than without one.

The validation and test data must be also normalized using the statistics (μ and σ) that were computed from the training data. Since a neural network is created for each desired output there is

no need to standardize the output. The output standardization is usually used if there are more than one output and they are in different order, hence affecting the way weights converge during the learning process. The statistical data is then collected to calculate root mean square error (RMSE) and coefficient of regression (R^2) values for both sets. Since there may be numerous metamodels created from the same sample set. RMSE and R^2 are the metrics used for goodness of fit. The created model may fit perfectly the training data set even though it may not qualify as a good model to represent the output for the given process at other points. Hence the verification data set is created so that the points are at the different locations than the sample data. If the verification dataset RMSE and R^2 values do not differ much from the training values, then the model has trained correctly, otherwise it must have been overfitted or trained improperly. If the neural model did not train correctly, the training parameters of the model can be adjusted or additional sample points can be collected from the circuit simulation. Otherwise the neural network can act as an accurate metamodel for the PLL circuit. A total of 200 simulations were needed to create the metamodels.

4.2 Proposed Method for Statistical Process Variation Analysis

In a typical approach, Monte Carlo (or its derivatives) are used *on the actual circuits (i.e. netlists)*. This is prohibitively slow. In our approach, the non-polynomial metamodels are used instead, thus speeding up significantly the statistical process variation analysis.

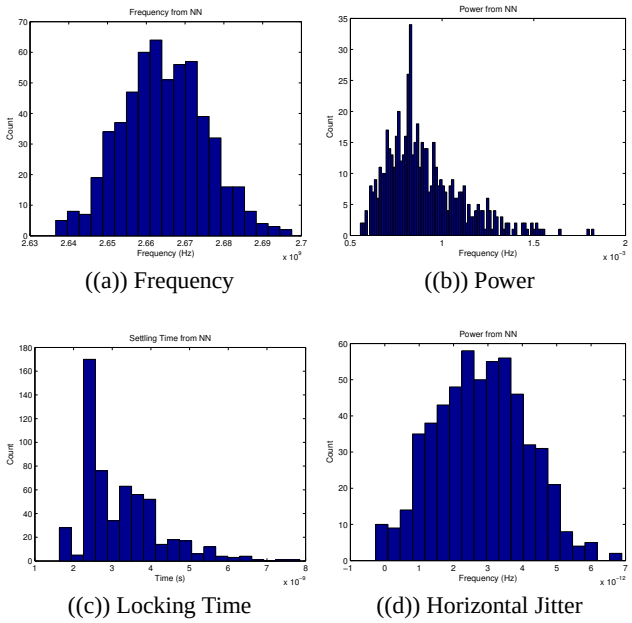
A Monte Carlo (MC) analysis of the PLL circuit containing full-blown parasitics is performed first for comparison purposes. The mean (μ) and standard deviations (σ) of the 4 Figures-of-Merit (FoMs) are presented in Table 1. It may be noted that **the processing time for 1000 MC runs on the actual circuit (i.e. netlist) was approximately 5 days**.

In the current non-polynomial metamodel-based design flow, the process variation analysis is conducted over the metamodels. The MC analysis is performed on the metamodels that show the best fit. The results are compared with the 1000 circuit Monte Carlo simulations. The PDFs extracted from the Monte Carlo analysis on the neural networks is shown in Fig. 2. The statistical parameters of these PDFs are also presented in Table 1.

5. PROPOSED PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

The PSO algorithm, uses multiple particles to find a solution based on the cost function. The particle movement is calculated based on the local intelligence of each particle which is offset using global knowledge. The steps of the approach are shown in Algorithm (1). Each particle location information holds a 35 dimensional location, where each dimension corresponds to a parameter. The algorithm starts at a random location of each parameter for each particle, with random velocity. When the information is acquired from the cost function $f(p_i)$ the minimum global position is retained in $f(g)$, while the local position is saved for each particle in p_i. With each iteration of the loop, while the amount of iteration is not reached, the particles keep searching for a minimum solution in the design space by updating the particle velocity vector v_i.

For the calculation of cost function, the Monte Carlo analysis is done around the x_i parameter points. To minimize the amount of calculations the analysis is only done on the frequency metamodel first. If the mean and standard deviation is within the specifications, then the rest of the metamodels are used to calculate the mean and standard deviation for other FoMs. Before the calculation of the cost value is done, all the values are brought to the same power to even their weight composition on the circuit.



((a)) Frequency ((b)) Power

((c)) Locking Time ((d)) Horizontal Jitter

Figure 2: Statistical analysis of the FoMs of the PLL using the *neural network metamodels*.

6. RESULTS AND CONCLUSIONS

PSO is used for optimization using neural network metamodels. Two optimization cases are considered in this paper. The first case optimizes the average case, while the worst case scenario is considered in the second case. A total of 35 parameters were used for both optimization cases. For Monte Carlo analysis each parameter is varied by 5% around the mean for process variation. The optimization results are shown in Table 2.

In this paper it is shown that neural network based metamodels can closely follow the behavior of a circuit for process variation and global optimization. On an example PLL system, the models were generated from the physical layout netlist for 4 different components. An error under 2% has been observed in the models for process variation analysis for mean and standard deviation. The PSO algorithm has been successfully used to bring the circuit back to specifications, even though the physical layout did not meet them. The processing time for running 1000 Monte Carlo analysis on the PLL system was approximately 5 days, in comparison to 200 simulations needed to create the metamodels and running the algorithm. The proposed design flow has reached a speedup of roughly 5 $\times$ over the Monte Carlo analysis without any iterative modifications to the physical layout.

Acknowledgements

This research is supported in part by SRC award P10883 and NSF awards CNS-0854182 and DUE-0942629.

7. REFERENCES

[1] K. Bernstein, D. J. Frank, A. E. Gattiker, W. Haensch, B. L. Ji, S. R. Nassif, E. J. Nowak, D. J. Pearson, and N. J. Rohrer. High-Performance CMOS Variability in the 65nm Regime

Table 1: Before Optimization: Statistical Figures of Merit of the PLL.

	Circuit Monte Carlo		Neural Network Monte Carlo		Error	
	Mean (μ)	Standard Deviation (σ)	Mean (μ)	Standard Deviation (σ)	Mean (μ)	Standard Deviation (σ)
Frequency	2.66 GHz	10.95 MHz	2.66 GHz	10.96 MHz	0.0%	0.11%
Power	0.90 mW	0.21 mW	0.90 mW	0.21 mW	0.14%	1.3%
Locking Time	3.24 μs	1.07 μs	3.22 μs	0.99 μs	0.7%	6.93%
Horizontal Jitter	2.79 ps	1.32 ps	2.80 ps	1.32 ps	0.12%	0.5%

Table 2: After Optimization: Statistical Figures of Merit of the PLL.

	$\mu + \sigma$ Optimization		$\mu + 3\sigma$ Optimization	
	Mean (μ)	Standard Deviation (σ)	Mean (μ)	Standard Deviation (σ)
Frequency	2.75 GHz	28.64 MHz	2.74 GHz	29.14 MHz
Power	0.99 mW	0.28 mW	0.98 mW	0.27 mW
Locking Time	4.69 μs	1.15 μs	4.61 μs	1.13 μs
Horizontal Jitter	5.82 ps	3.42 ps	5.97 ps	3.34 ps

Algorithm 1 The Proposed Particle Swarm Optimization (PSO) for the PLL Components.

Set N - number of particles
Start at a random location with uniform distribution
Get current position x_i and use it initially for best particle position $f(p_i)$ and $f(g) = min(p_i)$
$v_i\ U(min_{p\_i}, max_{p\_i})$
Initialize iter=0
Initialize weight for swarm effect ϱ_p
Initialize weight for swarm effect ϱ_p
Initialize weight for velocity effect (acceleration/inertia) w
while iter<$max_{iterations}$ **do**
  for each i **do**
    $v_i = \omega v_i + \varrho_p \tau_p (p_i - x_i) + \varrho_g \tau_g (g - x_i)$
    $x_i \leftarrow x_i + v_i$
    if $f(x_i) < f(p_i)$ **then**
      update position: $p_i \leftarrow x_i$
      if $f(p_i) < f(g)$ **then**
        $g \leftarrow p_i$
      end if
    end if
  end for
end while

and Beyond. *IBM Journal of Research and Development*, 50(4/5):433–449, July-Sep 2006.

[2] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De. Parameter Variations and Impact on Circuits and Microarchitecture. In *Proceedings of the 40th Annual Design Automation Conference*, DAC '03, pages 338–342, 2003.

[3] K. Bowman, S. Duvall, and J. Meindl. Impact of Die-to-Die and Within-Die Parameter Fluctuations on the Maximum Clock Frequency Distribution for Gigascale Integration. *IEEE Journal of Solid-State Circuits*, 37(2):183–190, February 2002.

[4] H.-S. Jeon, D.-H. You, and I.-C. Park. Fast frequency acquisition all-digital PLL using PVT calibration. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 2625–2628, 2008.

[5] C.-C. Kuo, M.-J. Lee, C.-N. Liu, and C.-J. Huang. Fast Statistical Analysis of Process Variation Effects Using Accurate PLL Behavioral Models. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 56(6):1160–1172, June 2009.

[6] J. Lin, B. Haroun, T. Foo, J.-S. Wang, B. Helmick, S. Randall, T. Mayhugh, C. Barr, and J. Kirkpatric. A pvt tolerant 0.18mhz to 600mhz self-calibrated digital pll in 90nm cmos process. In *Proceedings of the IEEE International Solid-State Circuits Conference*, pages 488–541, 2004.

[7] D. Miyashita, H. Ishikuro, S. Kousai, H. Kobayashi, H. Majima, K. Agawa, and M. Hamada. A phase noise minimization of cmos vcos over wide tuning range and large pvt variations. In *Proceedings of the IEEE Custom Integrated Circuits Conference*, pages 583–586, 2005.

[8] J. Park, K. Choi, and D. Allstot. Parasitic-aware Design and Optimization of a Fully Integrated CMOS Wideband Amplifier. In *Asia and South Pacific Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003.*, pages 904–907, Jan. 2003.

[9] R. Wu, J.-C. Wang, K.-W. Xia, and R.-X. Yang. Optimal Design on CMOS Operational Amplifier with QPSO Algorithm. In *International Conference on Wavelet Analysis and Pattern Recognition, 2008. ICWAPR '08.*, volume 2, pages 821 –825, Aug 2008.

[10] Y. Yang, L. Yang, and Z. Gao. A PVT Tolerant sub-mA PLL in 65nm CMOS Process. In *Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems*, pages 998–1001, 2008.