Ultra-Fast Design Exploration of Nanoscale Circuits through Metamodeling

Presenter:

Saraju P. Mohanty
NanoSystem Design Laboratory (NSDL)
Dept. of Computer Science and Engineering
University of North Texas, Denton, TX 76203, USA.
Email: saraju.mohanty@unt.edu

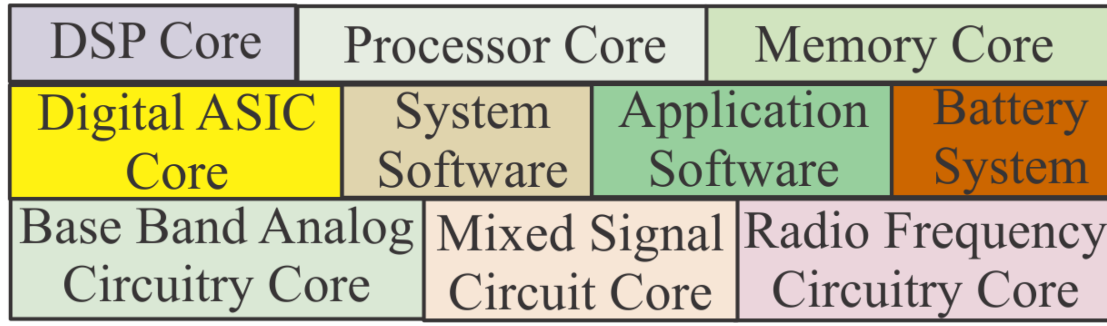
Outline of the Talk

- Nanoscale Design Challenges
- The Proposed Ultra-Fast Solution
- Metamodel Types and Proposed Techniques
- Algorithms for Optimization over Metamodels
- Experiments Using Case Studies
- Conclusions and Future Research

Outline of the Talk

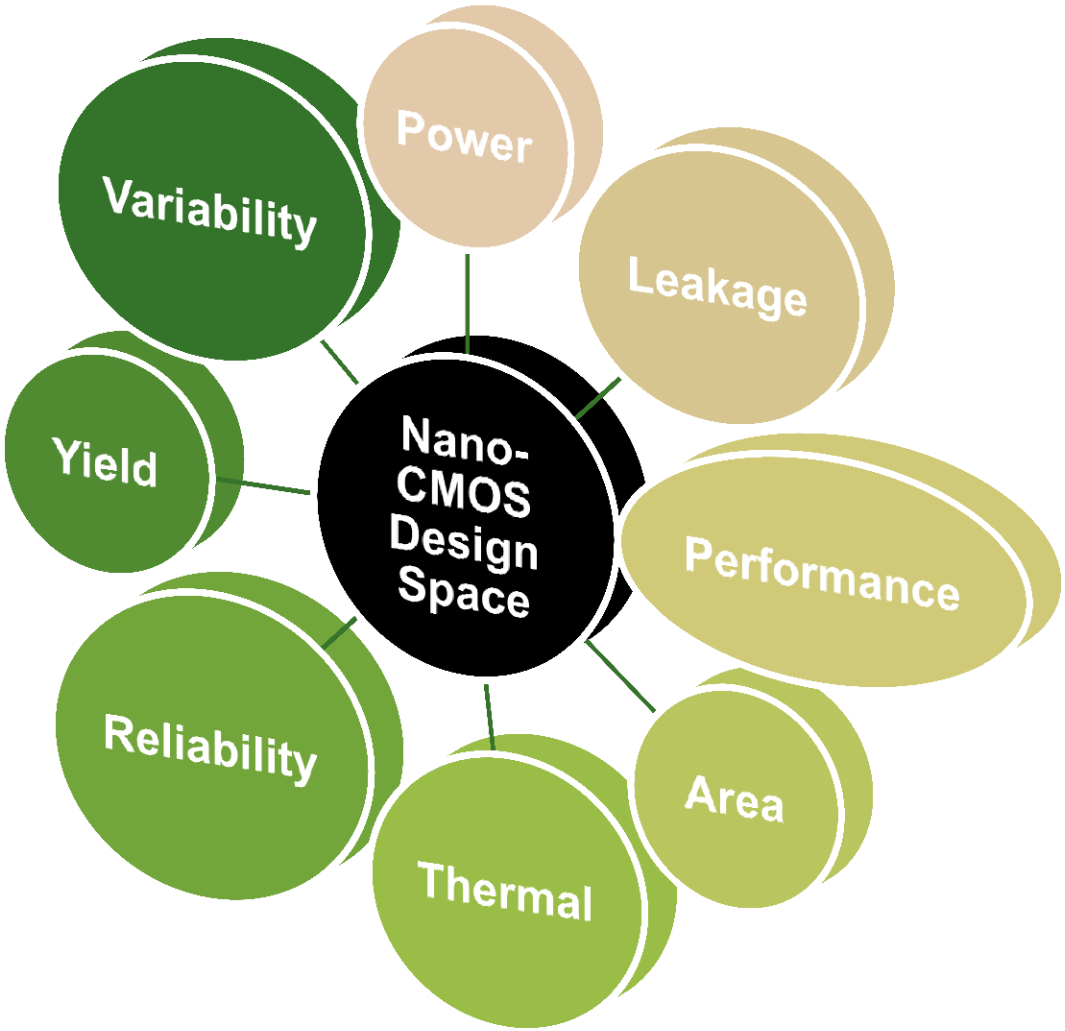
- **Nanoscale Design Challenges**
- The Proposed Ultra-Fast Solution
- Metamodel Types and Proposed Techniques
- Algorithms for Optimization over Metamodels
- Experiments Using Case Studies
- Conclusions and Future Research

Analog/Mixed-Signal Systems

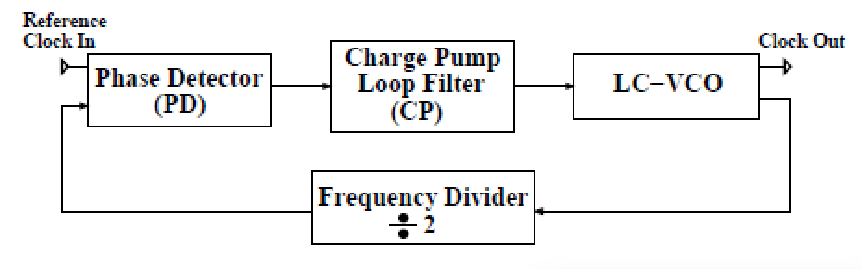


- A typical consumer electronics is an Analog/Mixed-Signal System-on-a-Chip (AMS-SoC).
- Individual subsystems can also be mixed-signal, e.g. Phase-Locked Loop (PLL).

Nano-CMOS Circuit: Design Space

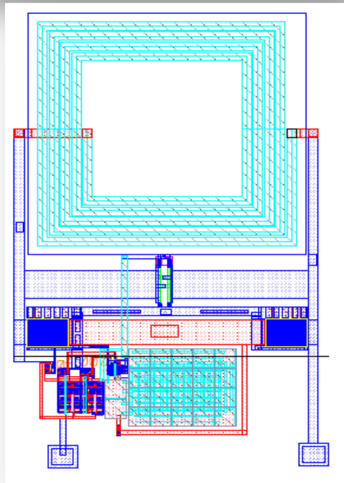


One of the Key Issues: Time/Effort



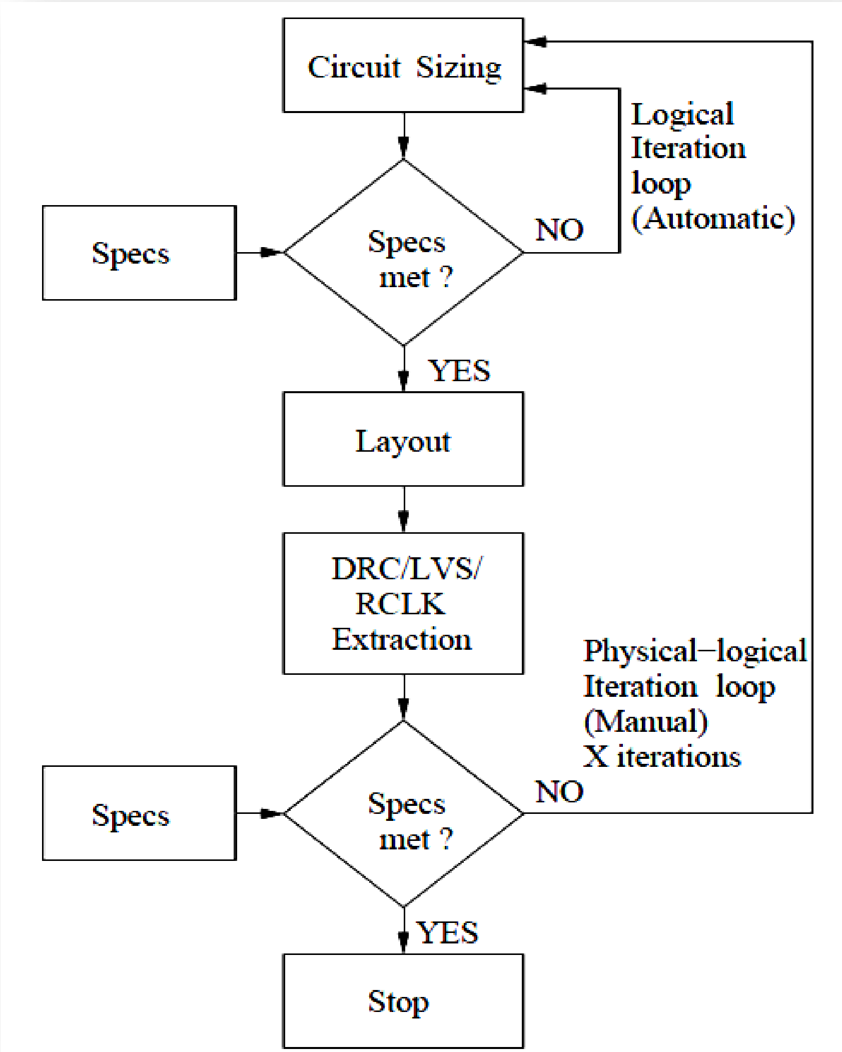
- The simulation time for a Phase-Locked-Loop (PLL) lock on a full-blown (RCLK) parasitic netlist is of the **order of many days**!

PLL



- **Issues for AMS-SoC components:**
  - How fast can design space exploration be performed?
  - How fast can layout generation and optimization be performed?

Standard Design Flow – Very Slow

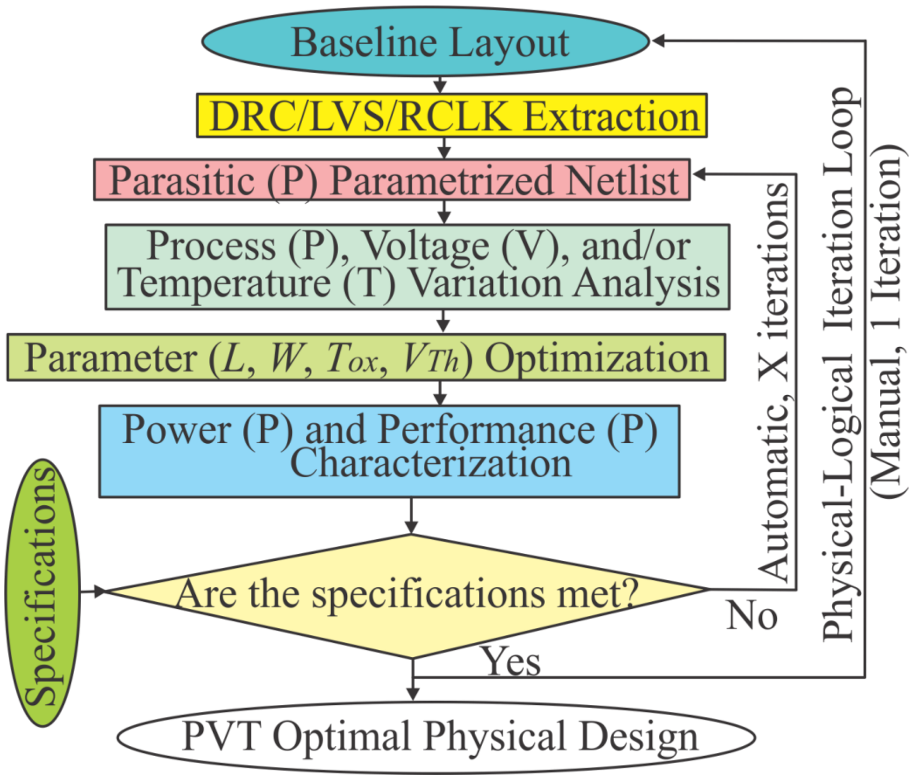


- Standard design flow requires multiple manual iterations on the back-end layout to achieve parasitic closure between front-end circuit and back-end layout.
- Longer design cycle time.
- Error prone design.
- Higher non-recurrent cost.
- Difficult to handle nanoscale challenges.

Outline of the Talk

- Nanoscale Design Challenges
- **The Proposed Ultra-Fast Solution**
- Metamodel Types and Proposed Techniques
- Algorithms for Optimization over Metamodels
- Experiments Using Case Studies
- Conclusions and Future Research

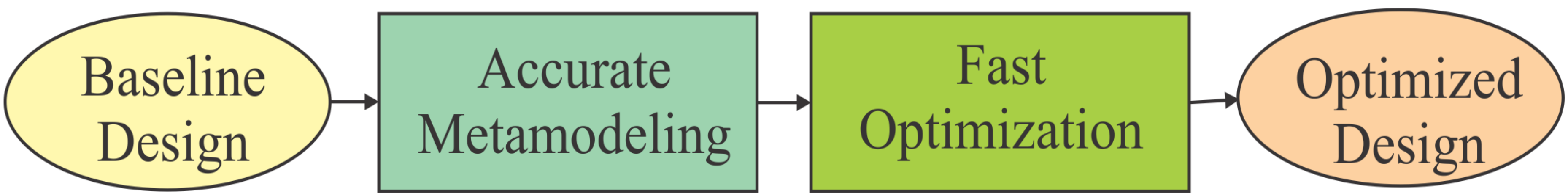
Automatic Optimization on Netlist (Faster than manual flow; still slow)



- Automatic iteration over netlist improves design optimization.
- Still needs multiple simulations using analog simulator (SPICE).
- SPICE is slow.

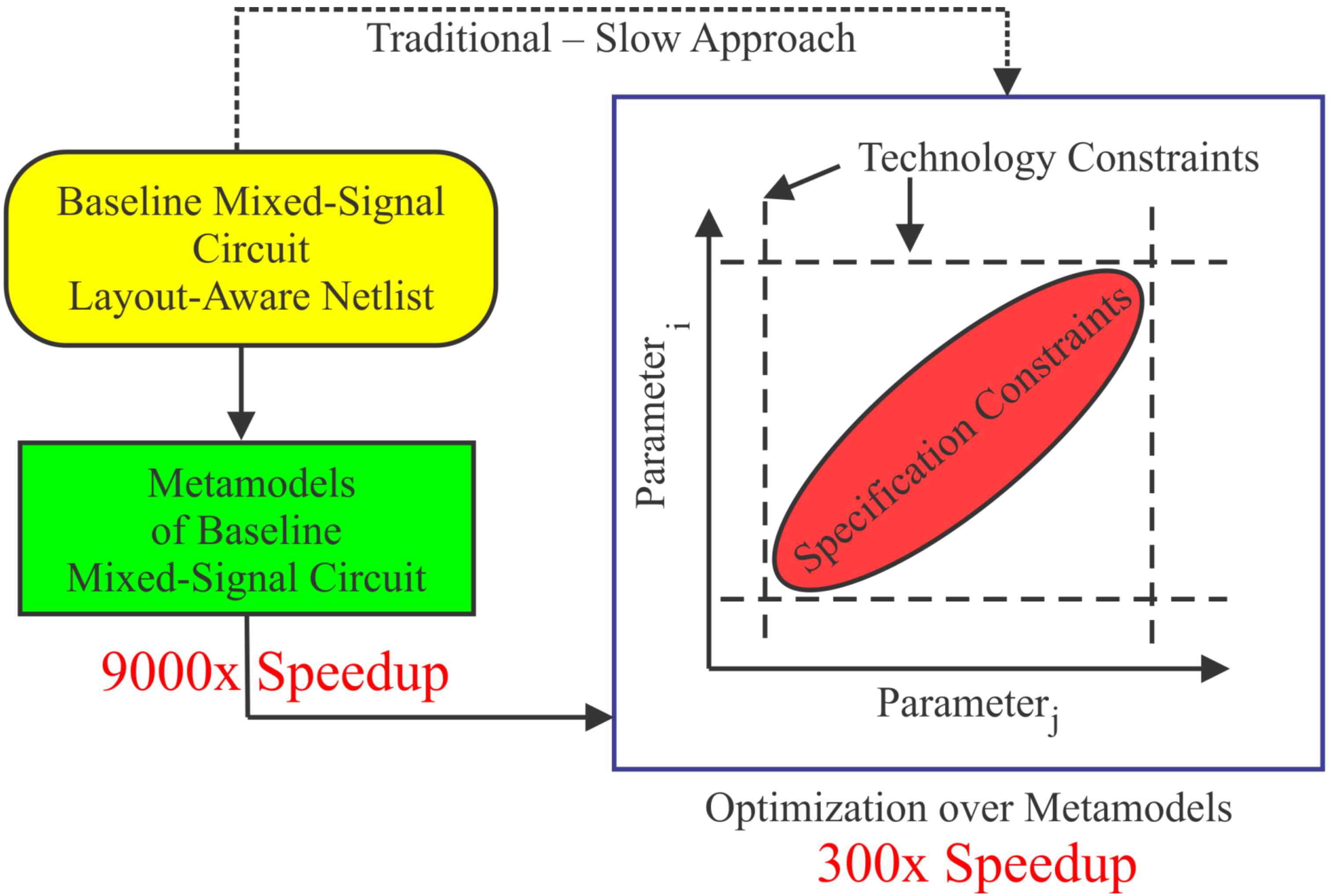
Ultra-Fast Design Exploration Through Metamodeling

The Actual Circuit (Netlist) Optimization -- Slow Approach



The Metamodel-Based Approach -- Ultra-Fast Approach

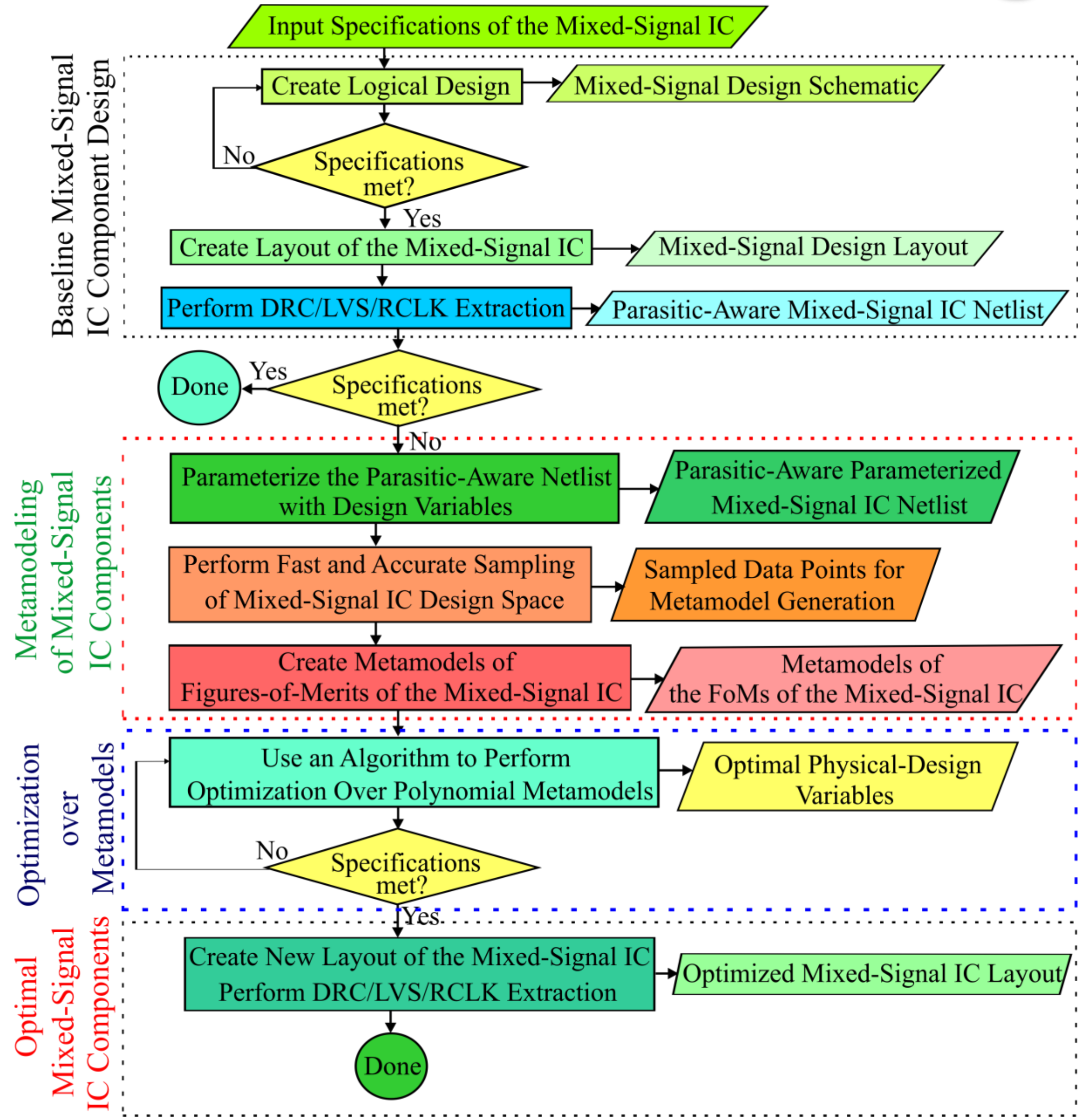
Two Tier Speed Up



Proposed Flow: Key Perspective

- Novel design and optimization methodology that will produce robust AMS-SoC components using **ultra-fast automatic iterations over metamodels** (instead of netlist) and two manual layout steps.
- The methodology easily accommodates multidimensional challenges, reduces design cycle time, improves circuit yield, and reduces chip cost.

Metamodel-Based Design Flow



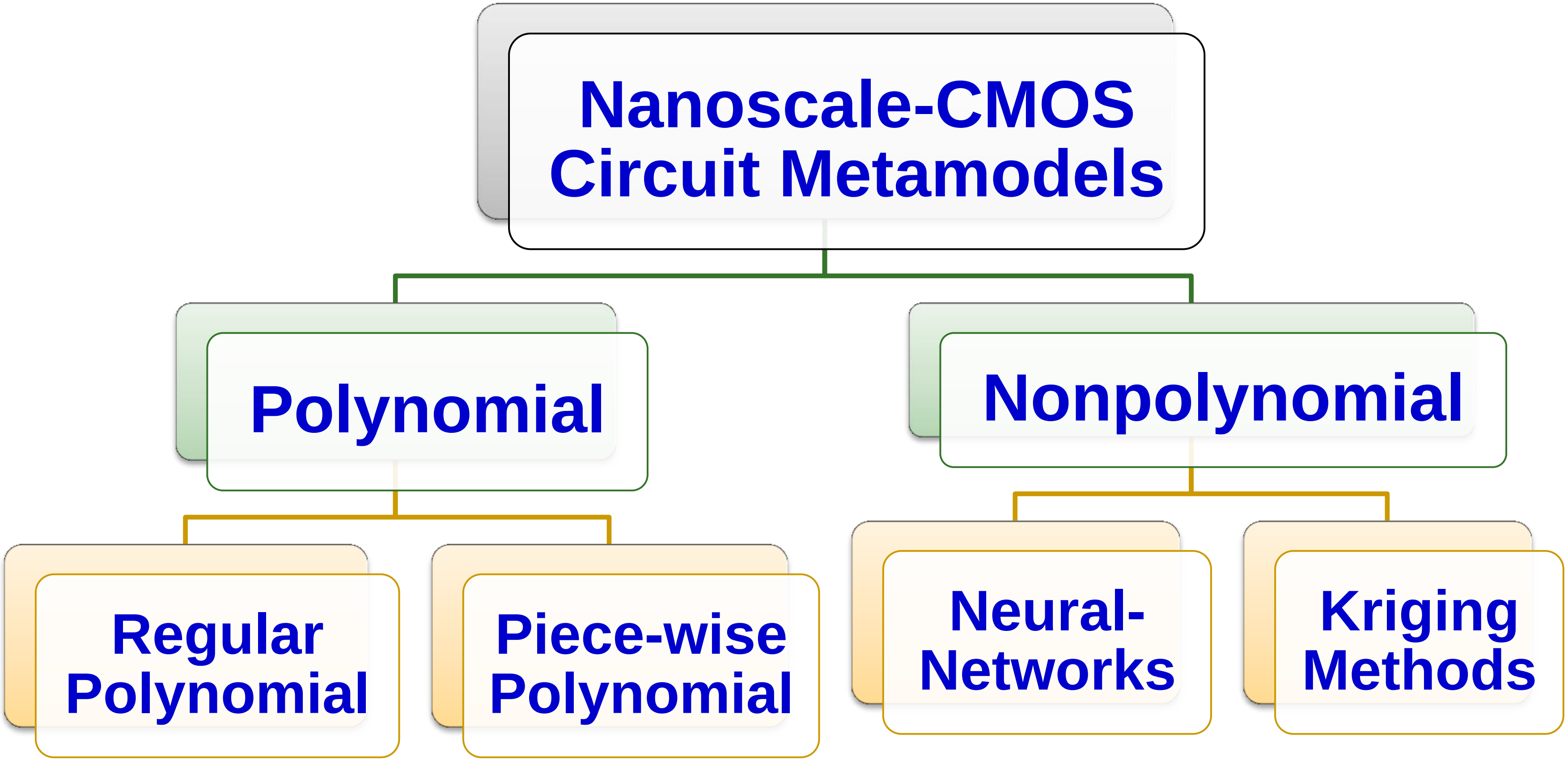
Metamodeling vs. Macromodeling

- **Macromodeling**
  - Simplified version of the circuit.
  - Used in the same simulation tool.
  - Hard to create.
- **Metamodeling**
  - Mathematical representation of output.
  - Based on prediction equation or algorithm.
  - Language and tool independent.
  - Reusable for different specifications.
  - Can be applied using non-EAD tools like MATLAB.

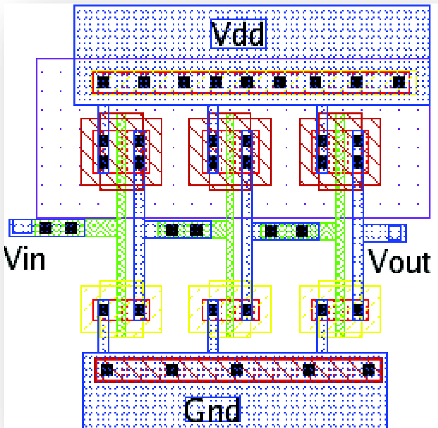
Outline of the Talk

- Nanoscale Design Challenges
- The Proposed Ultra-Fast Solution
- **Metamodel Types and Proposed Techniques**
- Algorithms for Optimization over Metamodels
- Experiments Using Case Studies
- Conclusions and Future Research

Metamodels : Selected Types

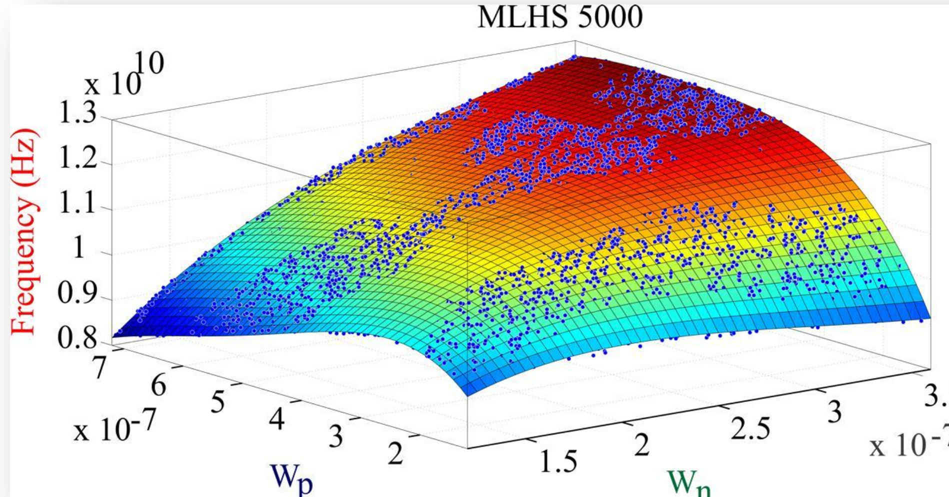


Metamodels : Polynomial Example



Actual Circuit (SPICE netlist) of AMS-SoC Components

Statistical Sampling



Polynomial Function Fitting

$$f(W_n, W_p) = 7.94 \times 10^9 + 1.1 \times 10^{16} W_n + 1.28 \times 10^{15} W_p.$$

Metamodeling – Key Points

- **Accuracy** -- Capability of predicting the system response over the design space.
- **Efficiency** -- Computational effort required for constructing the metamodel.
- **Transparency** -- Capability of providing the information concerning contributions and variations of design variables and correlation among the variables.
- **Simplicity** -- Simple methods should require less user input and be easily adapted to different problem.

Metamodels: Performance Analysis

- **Root-Mean Square Error (RMSE)**: Represents departure of metamodel from real-simulation (golden). Smaller RMSE means more accurate:

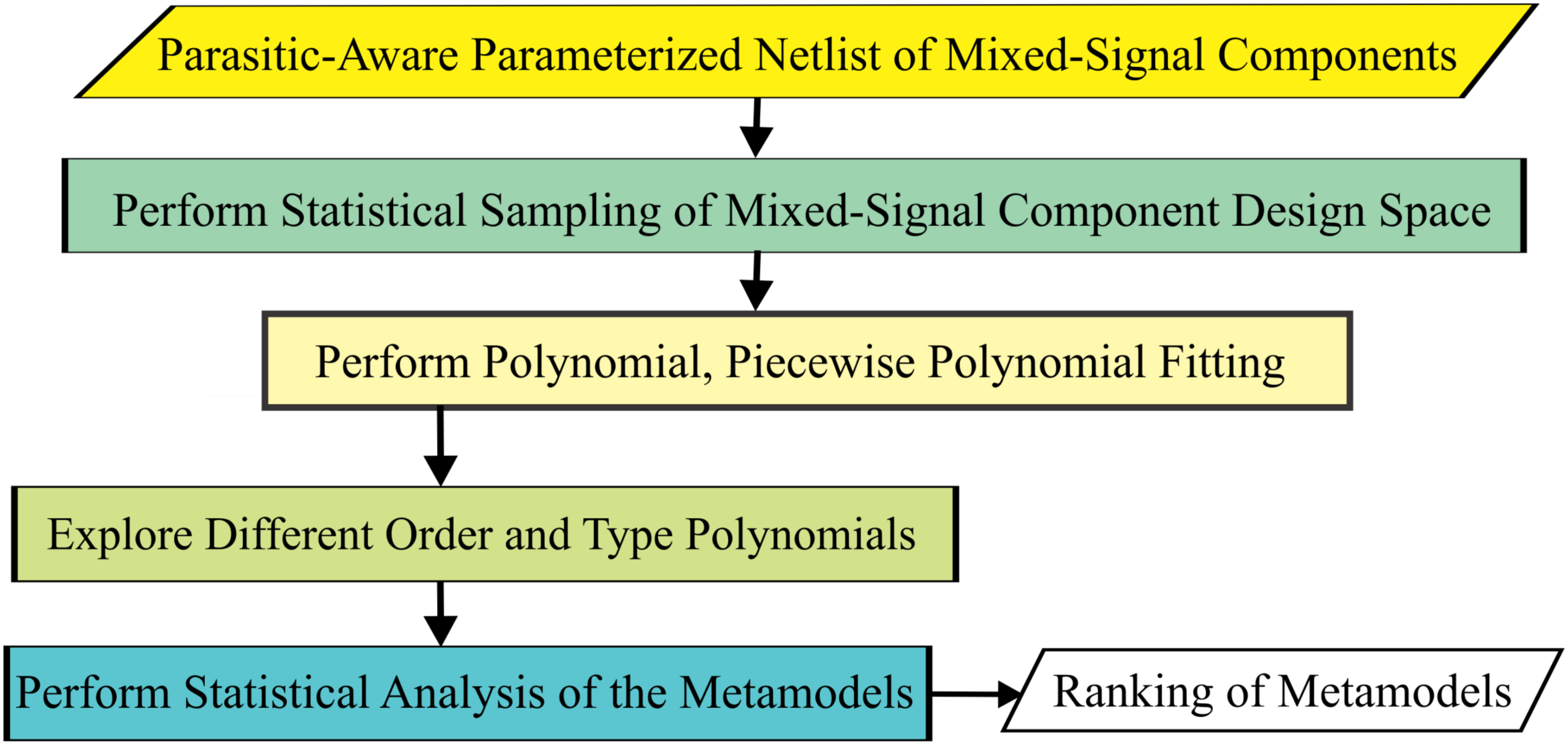
$$\mathrm{RM}SE = \sqrt{\left(\frac{1}{N}\right)\sum_{k=1}^{N}\left(FoM(x_k) - \widehat{FoM}(x_k)\right)^2}$$

- **Relative Average Absolute Error (RAAE)**: Smaller RAAE means more accurate metamodel:

$$RAAE = \left(\frac{\sum_{k=1}^{N}|FoM(x_k) - \widehat{FoM}(x_k)|}{N \times Standard\ Deviation}\right)$$

- **R-Square**: Larger R-square means more accurate metamodel: $R^2 = \left(1 - \frac{MSE}{Variance}\right)$

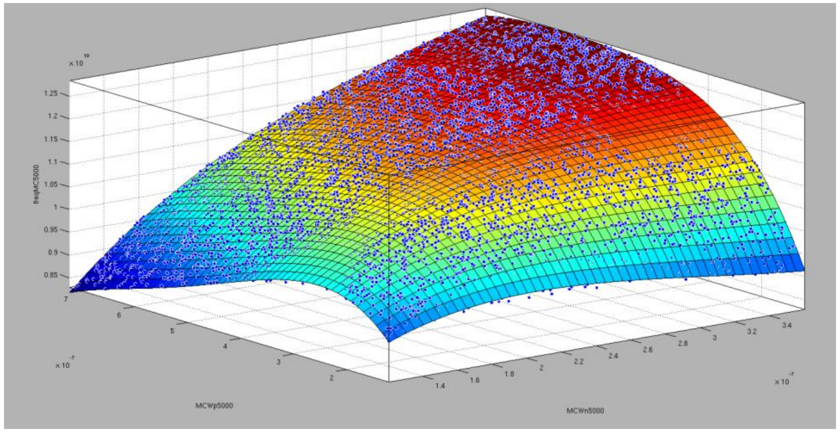
Metamodel Generation Flow



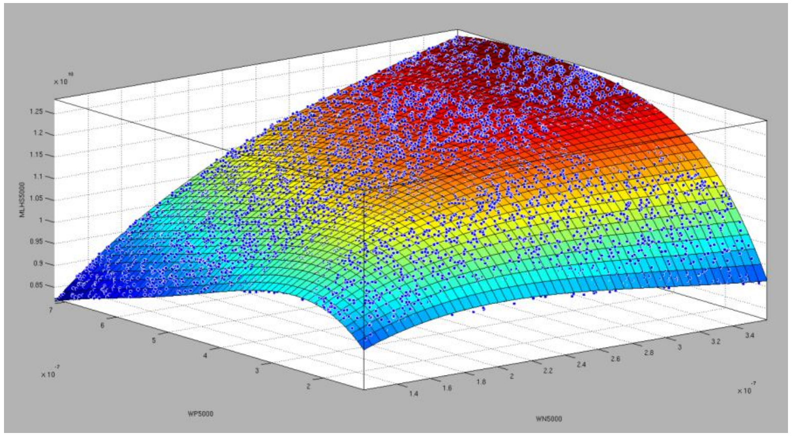
- Different flow is used for nonpolynomial metamodel generation.

Sampling Techniques: 45nm Ring Oscillator Circuit (5000 points)

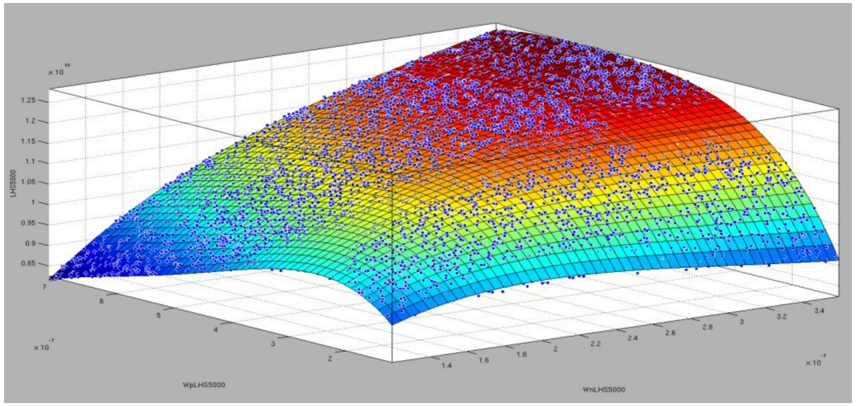
Monte Carlo



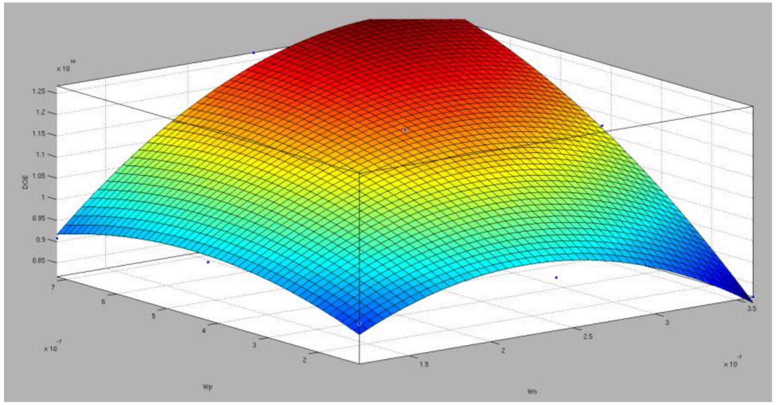
MLHS



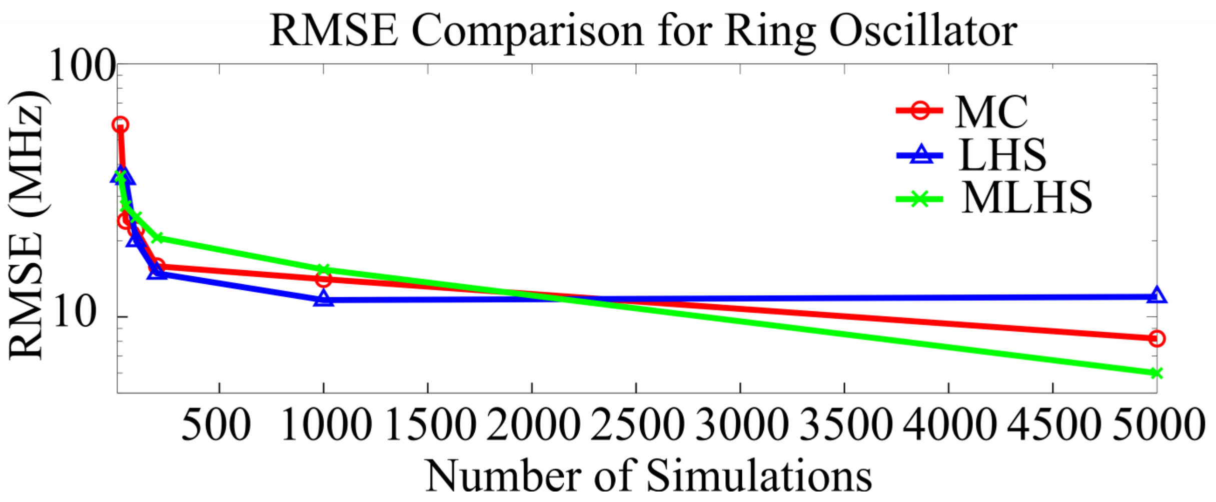
LHS

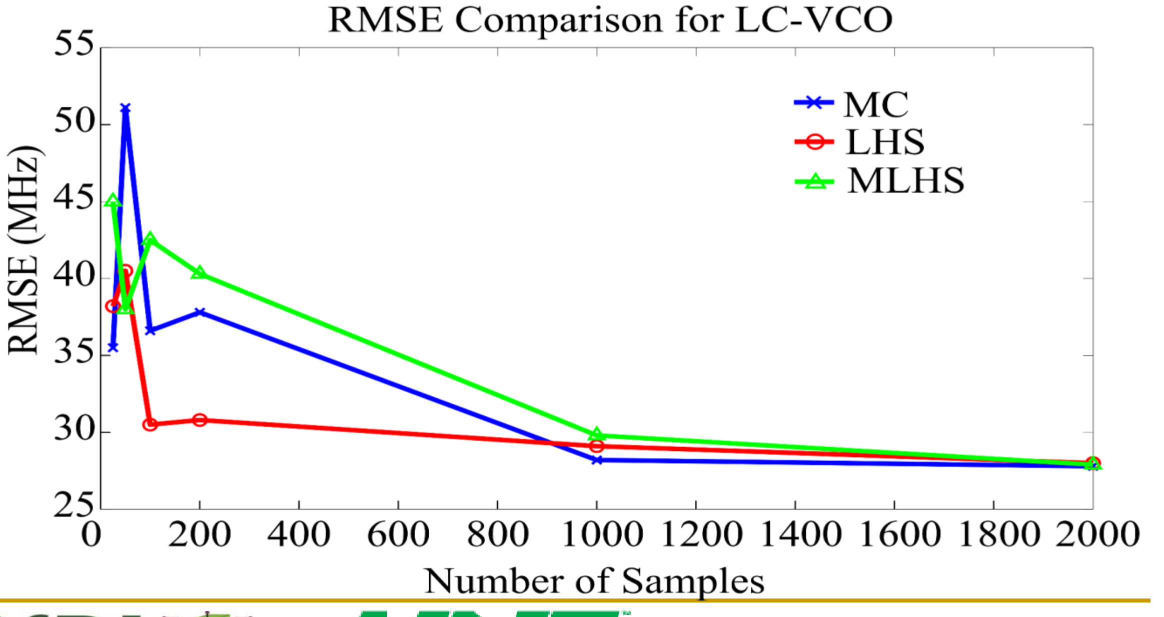


DOE



Sampling Comparison: RO / LC-VCO





Polynomial Metamodels

- The generated sample data can be fitted in many ways to generate a metamodel.
- The choice of fitting algorithm can affect the accuracy of the metamodel.
- A simple metamodel has the following form:

$$y = \sum_{i,j=0}^{k} \left(\alpha_{ij} \times x_1^i \times x_2^j \right)$$

- *y* is the response being modeled (e.g. frequency), $x = [W_n, W_p]$ is the vector of variables and α_{ij} are the coefficients.

Metamodel: Polynomial Comparison

Case Study Circuits	Polynomial Order	μ error (in MHz)	σ error (in MHz)
Ring Oscillator **45nm CMOS** **Target *f* : 10GHz**	1	571.0	286.7
	2	195.4	78.1
	3	37.2	18.0
	4	20.0	10.7
	5	17.1	9.6
LC-VCO **180nm CMOS** **Target *f* : 2.7GHz**	1	42.3	40.1
	2	39.4	37.8
	3	35.4	33.9
	4	30.5	29.3
	5	26.5	25.2

Ring oscillator – Order 1

$$f(W_n, W_p) = 7.94 \times 10^9 + 1.1 \times 10^{16} W_n + 1.28 \times 10^{15} W_p.$$

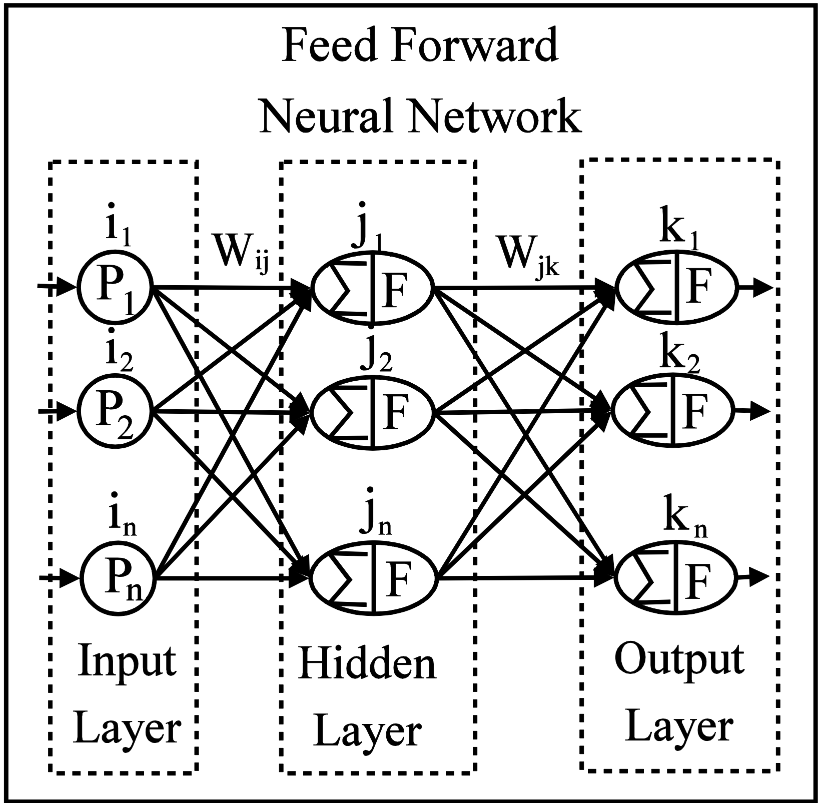
LC-VCO – Order 1

$$f(W_n, W_p) = 2.38 \times 10^9 - 3.49 \times 10^{12} W_n - 6.66 \times 10^{12} W_p.$$

Neural Network Metamodeling

- Feed-forward dual layer NNs (FFDL) are considered.
- FFDL network created for each FoM:
  - Nonlinear hidden layer functions are considered each varying hidden neurons 1-20:

$$b_j\,(v_j) = \tanh(\lambda v_j)$$



Metamodel Comparison: Polynomial Vs Nonpolynomial

- Nonpolynomial (Neural Network) is more suitable large circuits.

180nm CMOS PLL with Target Specs: f = 2.7GHz, P = 3.9mW, 8.5μs.

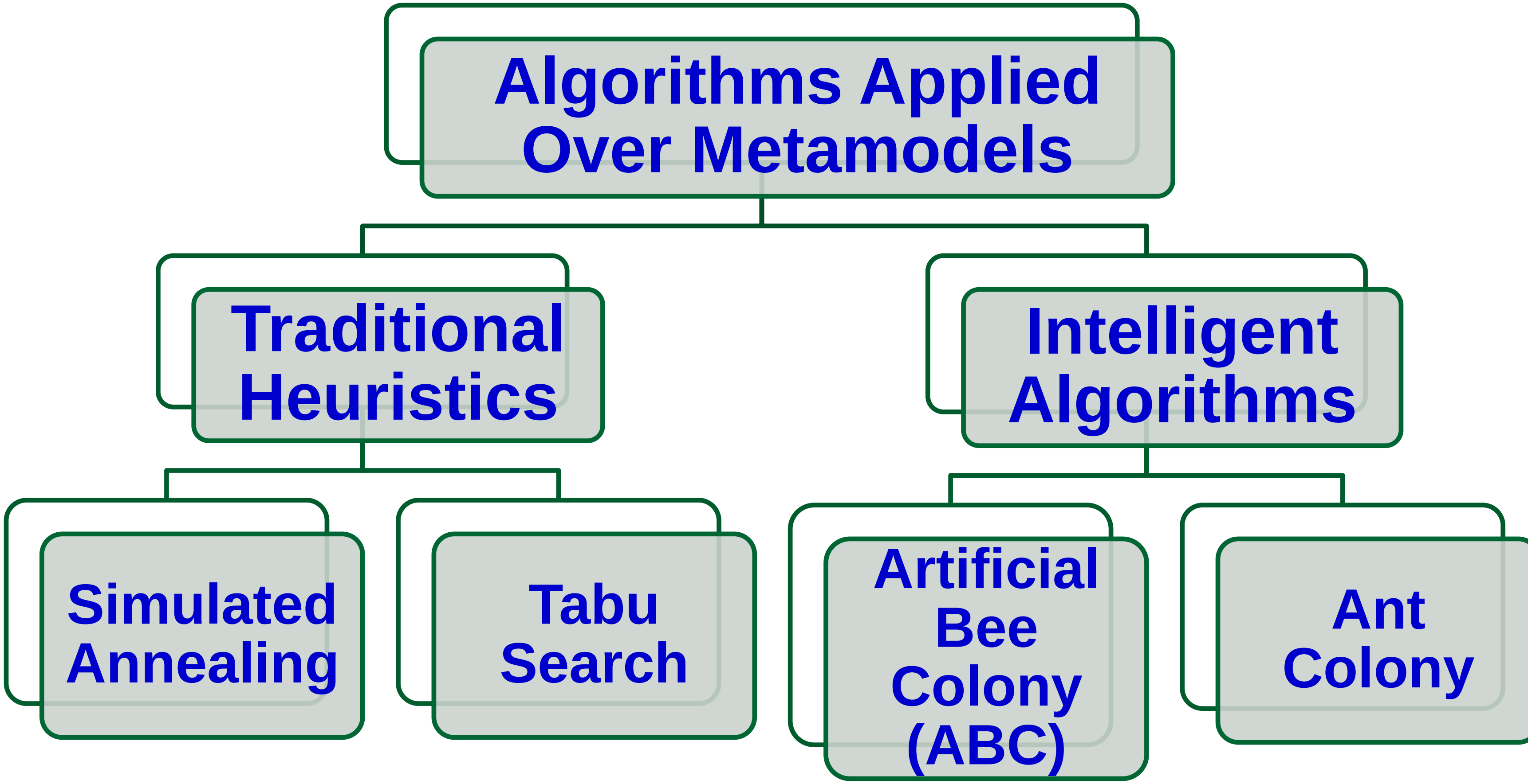
Figures-of-Merits (FoM)	Polynomial		Nonpolynomial (Neural Network)
	# of Coefficients	RMSE	
Frequency	48	77.96 MHz	48MHz
Power	50	2.6mW	0.29mW
Locking Time	56	1.9μs	1.2μs

- 56% increase in accuracy over polynomial metamodels.
- On average 3.2% error over golden design surface.

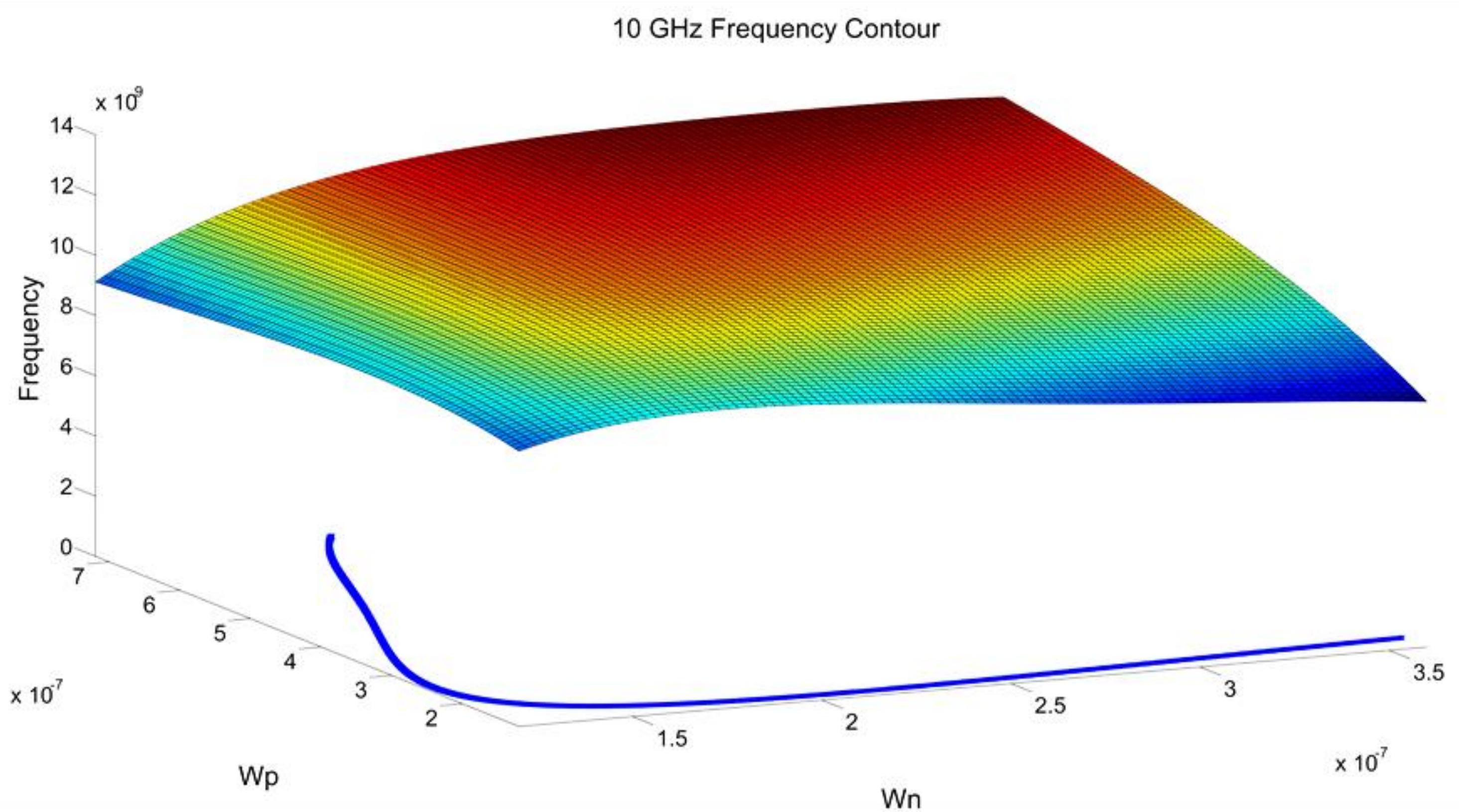
Outline of the Talk

- Nanoscale Design Challenges
- The Proposed Ultra-Fast Solution
- Metamodel Types and Proposed Techniques
- **Algorithms for Optimization over Metamodels**
- Experiments Using Case Studies
- Conclusions and Future Research

Selected Algorithms for Optimization over Metamodels

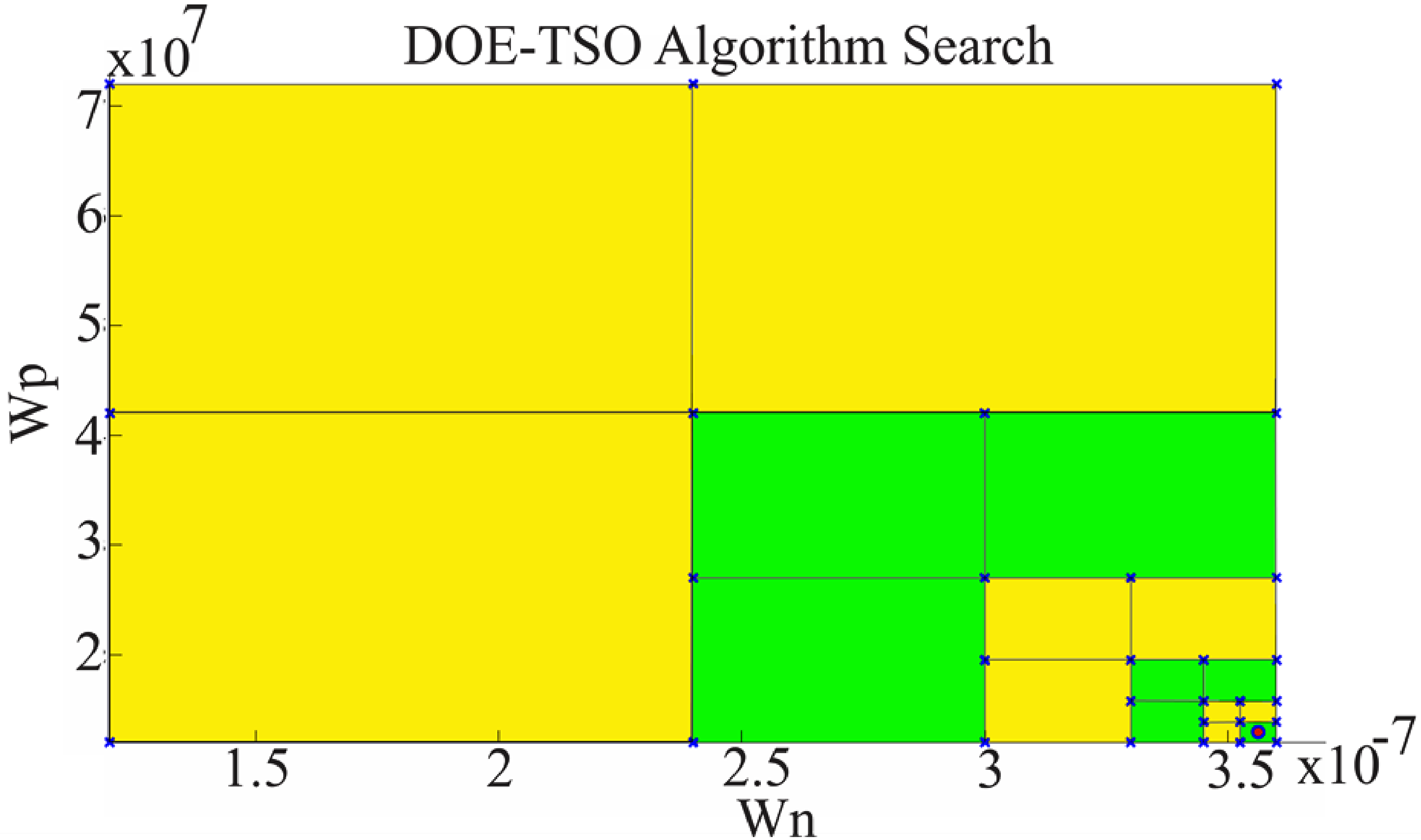


Exhaustive Search : 45nm RO



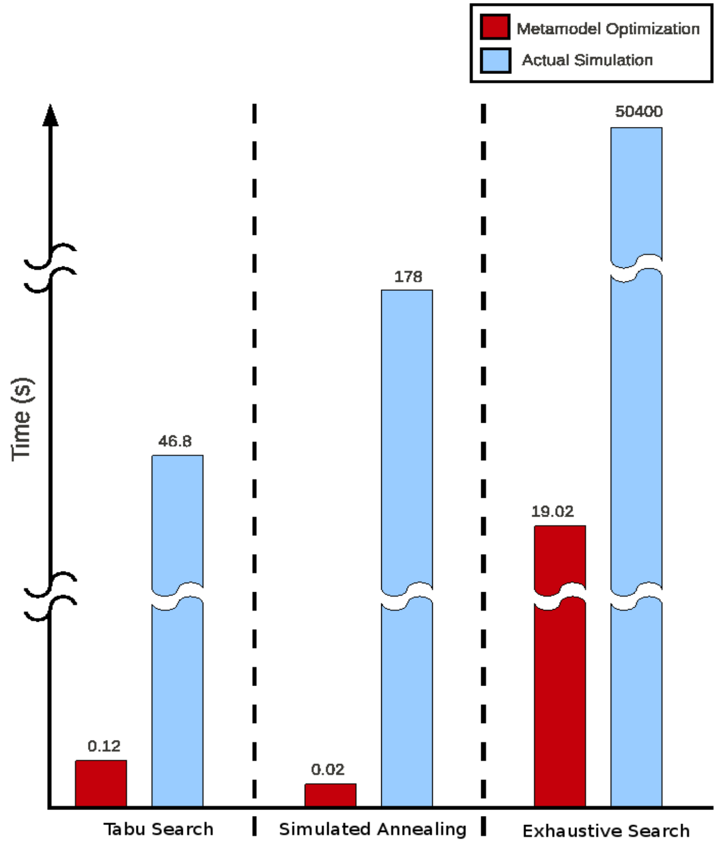
- Searches over two parameter space.
- Parameters incremented over specified steps.

DOE Assisted Tabu Search: 45nm RO



- Search space is recursively divided into rectangles and each time the rectangle with superior result is selected.

Comparison of the Running Time of Heuristic Algorithms: 45nm RO



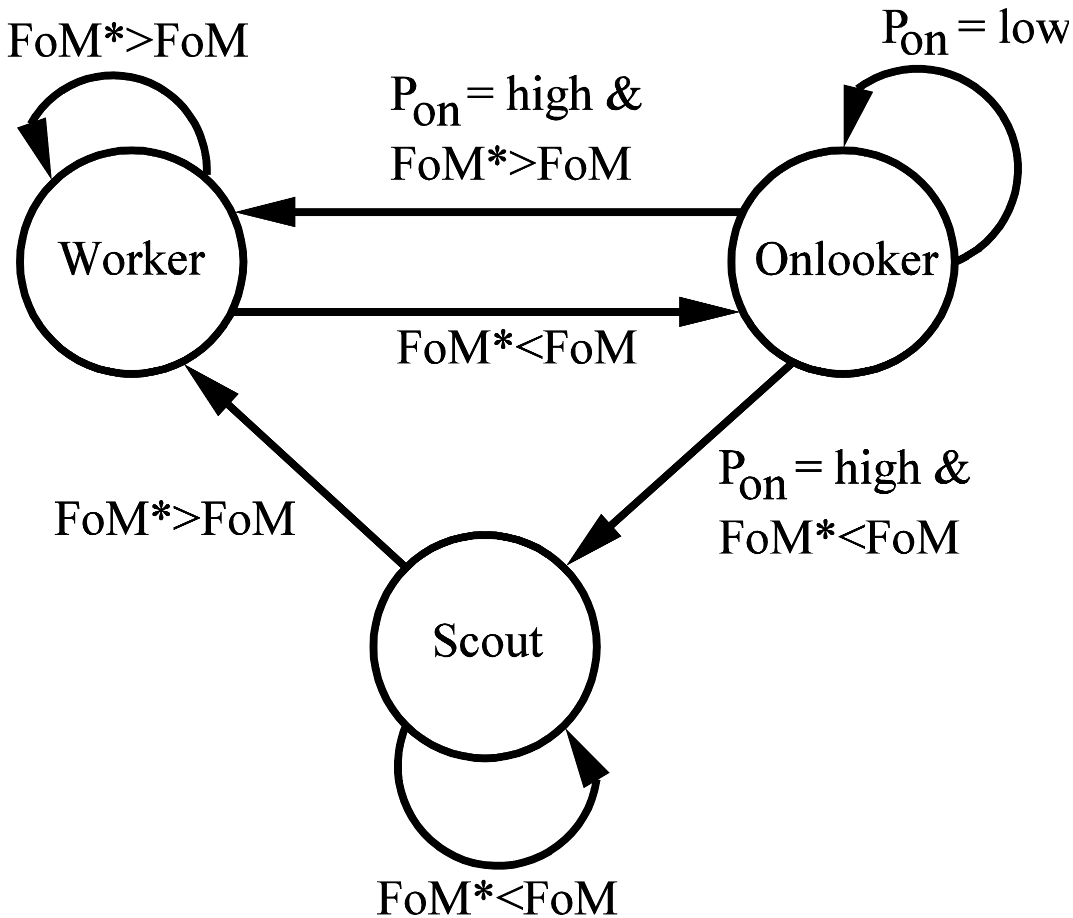
- **Optimization without metamodels:** the tabu search optimization is faster by ~1000× than the exhaustive search and ~4× faster than the simulated annealing optimization.
- **Optimization with metamodels:** the simulated annealing optimization is faster by ~1000× than the exhaustive search and ~6× faster than the tabu search optimization.

Bee-Colony Optimization: Overview

1. **Initial** food sources are produced for all worker bees.
2. **Do**
   1) Each worker bee goes to a food source and evaluates its nectar amount.
   2) Each onlooker bee watches the dance of worker bees and chooses one of their sources depending on the dances and evaluates its nectar amount.
   3) Determine abandoned food sources and replace with the new food sources discovered by scout bees.
   4) Best food source determined so far is recorded.
3. **While** (requirements are met)

A food source → a solution; A position of a food source → a design variable set; Nectar amount → Quality of a solution; Number of worker bees → number of quality solutions.

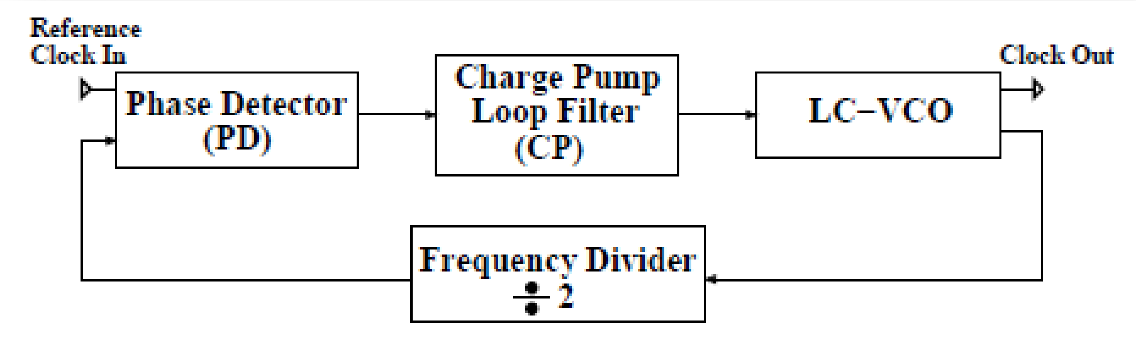
Bee Colony Optimization: States



Outline of the Talk

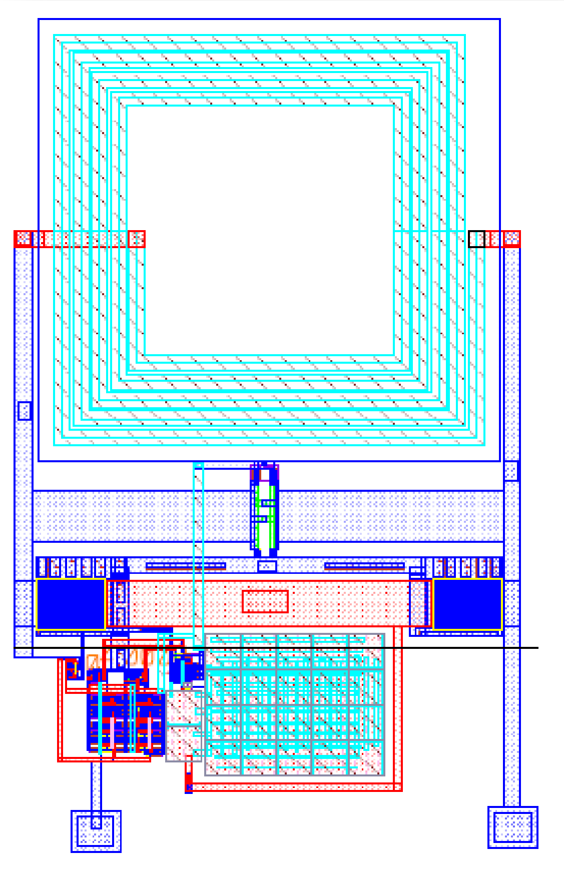
- Nanoscale Design Challenges
- The Proposed Ultra-Fast Solution
- Metamodel Types and Proposed Techniques
- Algorithms for Optimization over Metamodels
- **Experiments Using Case Studies**
- Conclusions and Future Research

Case Study Circuit: 180nm PLL



Block diagram of a PLL.

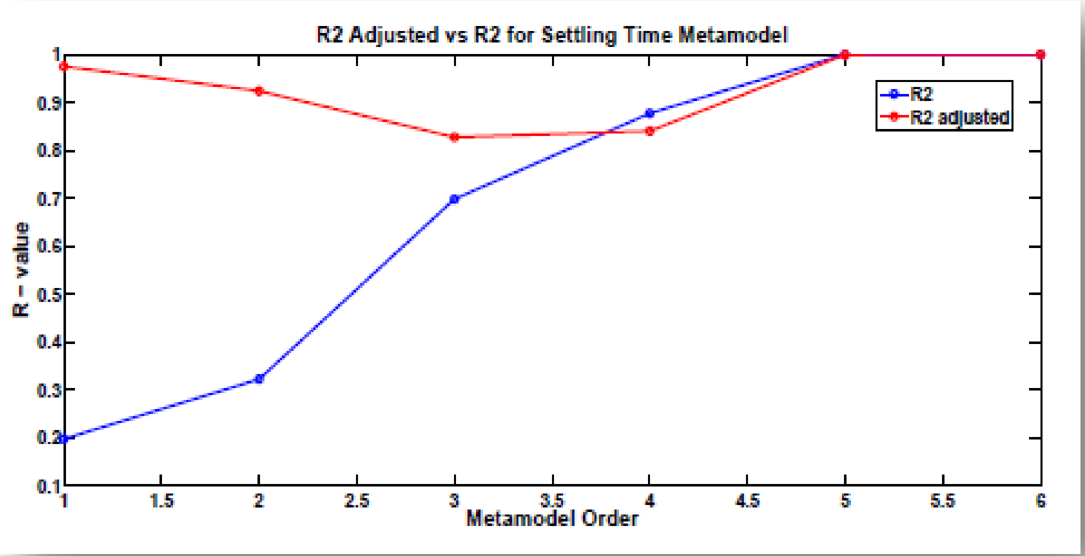
- PLL circuit is characterized for frequency, power, vertical and horizontal jitter (for simple phase noise), and locking time.
- Metamodels are created for each FoM from same sample set.



PLL for 180nm.

PLL: Polynomial Metamodels …

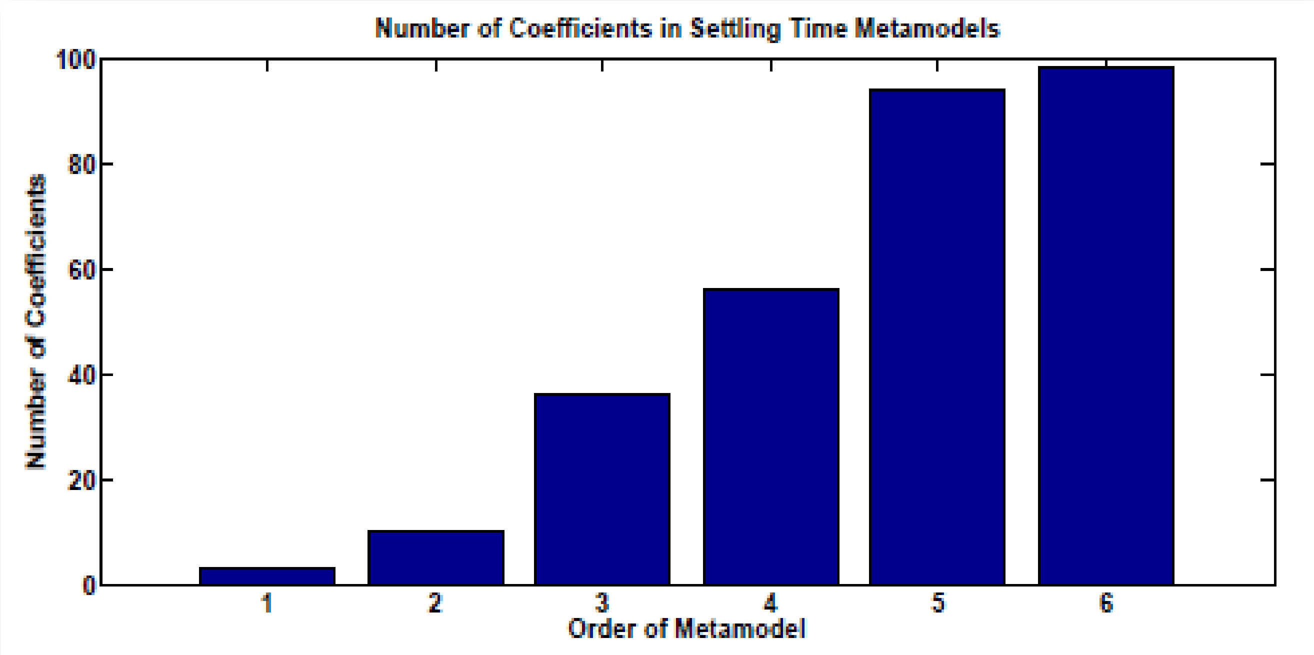
- PLL circuit is characterized for output frequency, power, vertical and horizontal jitter (to simplify the phase noise calculations), and locking time (or settling time).
- A separate metamodel is created for each FoM from the same sample set.
- The Root Mean Square Error (RMSE) and coefficient of determination R^2 are the metrics used for goodness of fit.



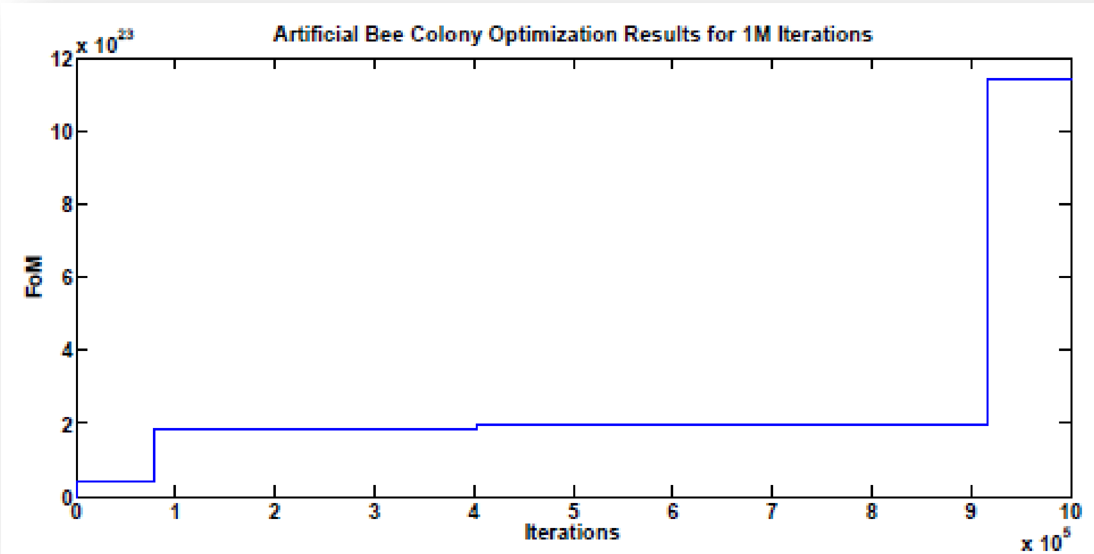
Generated R^2 and R^2_{adj} for various orders of the polynomial metamodel for settling time. Notice possible overfitting.

PLL: Polynomial Metamodels …

- The number of coefficients corresponding to the order of the generated metamodel for settling time.
- This means that the model is over fitted, therefore for the metamodel that represents settling time, a polynomial order of 4 will be used.



PLL: ABC over Poly. Metamodels …



The Artificial Bee-Colony (ABC) Optimization algorithm progression for the selected FoM.

Power and Jitter Results of the PLL

Metric	Before Optimization	After Optimization	Improvement
Power	9.29 mW	0.87 mW	90.6%
Jitter Vertical	168.35μV	3.28 nV	$\sim$100%
Jitter Horizontal	189 ps	180 ps	4.8%

PLL: ABC over Poly. Metamodels

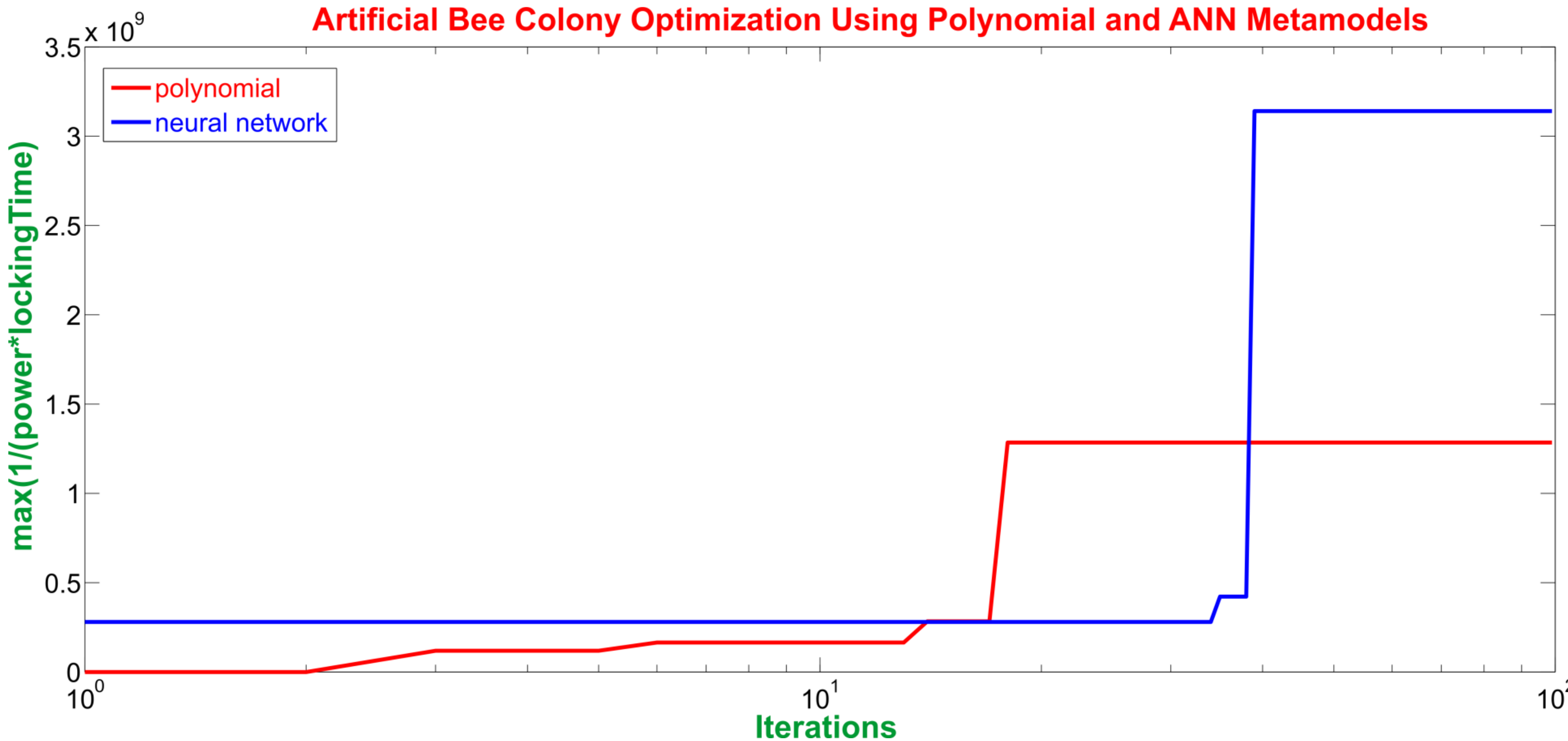
PLL parameters with constraints and optimized values.

Circuit	Parameter	Min (m)	Max (m)	Optimal Value (m)
Phase Detector	W_{ppd1}	400n	2μ	1.66μ
	W_{npd1}	400n	2μ	1.11μ
	W_{ppd2}	400n	2μ	784n
	W_{npd2}	400n	2μ	689n
	W_{ppd3}	400n	2μ	1.54μ
	W_{npd3}	400n	2μ	737n
Charge Pump	W_{nCP1}	400n	2μ	1.24μ
	W_{pCP1}	400n	2μ	1.35μ
	W_{nCP2}	1μ	4μ	1.35μ
	W_{pCP2}	1μ	4μ	2.88μ
LC-VCO	W_{nLC}	3μ	20μ	18.62μ
	W_{pLC}	6μ	40μ	37.48μ
Divider	W_{p1Div}	400n	2μ	1.65μ
	W_{p2Div}	400n	2μ	1.54μ
	W_{p3Div}	400n	2μ	1.38μ
	W_{p4Div}	400n	2μ	1.96μ
	W_{n1Div}	400n	2μ	1.09μ
	W_{n2Div}	400n	2μ	1.17μ
	W_{n3Div}	400n	2μ	1.29μ
	W_{n4Div}	400n	2μ	1.95μ
	W_{n5Div}	400n	2μ	536n

- An exhaustive search of the design space of 21 parameters with 10 intervals per parameter requires 10^{21} simulations.
- 10^{21} SPICE simulations is slow; 10min per one.
- 10^{21} simulations using polynomial metamodels is fast.
- Time savings: $\approx 10^{20}\times$ SPICE simulation time.

PLL: ABC Optimization: Poly. Vs NN

- Figure-of-Merit used for optimization objective function of PLL: $FoM = \left(\frac{1}{Power \times Locking\ Time}\right)$.



PLL: ABC Optimization: Poly. Vs NN

Optimization Results

FoM	Poly. Metamodel	ANN Metamodel
Average Power	3.9 mW	3.9 mW
Frequency	2.6909 GHz	2.7026 GHz

Optimization Time Comparison

Algorithm	Circuit Netlist	Poly. Metamodel	ANN Metamodel
ABC (100 iterations)	#bees(20) * 5 min * 100 iteration = 10,000 minutes = **7 days (worst case)**	5 mins	0.12 mins
Metamodel Generation	0	11 hours for LHS + 1 min creation	11 hours for LHS + 10mins training and verification.

Outline of the Talk

- Nanoscale Design Challenges
- The Proposed Ultra-Fast Solution
- Metamodel Types and Proposed Techniques
- Algorithms for Optimization over Metamodels
- Experiments Using Case Studies
- **Conclusions and Future Research**

Related Prior Research

Fast Design Exploration

- **Neural Networks**
  - Xia 2009
  - Wolfe 2003
  - Xu 2003
- **Intelligent Algorithms**
  - Delican 2010
  - Thakker 2009
  - Fernando 2008
- **Macromodeling**
  - Basu 2009
  - Ding 2006
  - Agarwal 2005
- **Metamodeling**
  - Samanta 2010
  - Lamecki 2008
  - Wong2006

Neural Networks, Intelligent Algorithms → **Actual SPICE netlist**

Macromodeling → **Simplified SPICE**

Metamodeling → **Functions**

Conclusions …

- Polynomial/nonpolynomial metamodels are explored.
- Use of metamodels and optimization algorithm speed up the design-space exploration for AMS circuits.
- LHS was identified as an accurate sampling method.
- Polynomial metamodels are easier create but can be applied for small circuits.
- 56% increase in accuracy is observed using feed forward NN over polynomial metamodels.
- On average 3.2% error is observed using NN.

Conclusions

- As a case study, a 180nm PLL, the circuit was parameterized with 21 parameters and optimized using the ABC algorithm.
- The final outcome of the design flow was 90% power savings and and average of 52% jitter minimization.
- Only 100 simulations are used to generate the accurate metamodels and ABC converged faster.
- An exhaustive search of the design space of 21 parameters with 10 intervals per parameter would require 10^{21} simulations. The time savings are enormous ($\approx 10^{20} \times$ SPICE simulation time).

Our Selected Publication on this Research

- 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 (TSM)*, Vol. 25, No. 1, February 2012, pp. 26--36.
- O. Garitselov, **S. P. Mohanty**, and E. Kougianos, “Fast-Accurate Non-Polynomial Metamodeling for nano-CMOS PLL Design Optimization”, in *Proceedings of the 25th IEEE International Conference on VLSI Design (VLSID)*, pp. 316—321.
- O. Garitselov, **S. P. Mohanty**, E. Kougianos, and O. Okobiah, "Metamodel-Assisted Ultra-Fast Memetic Optimization of a PLL for WiMax and MMDS Applications", in *Proc. 13th IEEE International Symposium on Quality Electronic Design (ISQED)*, pp. 580—585.
- O. Garitselov, **S. P. Mohanty**, and E. Kougianos, “Fast Optimization of Nano-CMOS Mixed-Signal Circuits Through Accurate Metamodeling”, in *Proceedings of the 12th IEEE International Symposium on Quality Electronic Design (ISQED)*, pp. 405--410, 2011.

Future Research

- Capturing statistical process variations using metamodels
- Kriging metamodeling
  - Effective handle correlations
  - Accurately model process variations
- Integration in HDLs
  - Used for accurate behavioral simulations
- Application to MEMS/NEMS
  - Unified simulation and design exploration of heterogeneous components

Thank you !!!