Fast Optimization of Nano-CMOS Mixed-Signal Circuits Through Accurate Metamodeling

Oleg Garitselov, Saraju P. Mohanty, Elias Kougianos
NanoSystem Design Laboratory (http://nsdl.cse.unt.edu), Dept. of Computer Science and Engineering
University of North Texas, USA. Email-ID: saraju.mohanty@unt.edu

Abstract

- *Design optimization methodologies for AMS-SoCs with analog, digital and mixed-signal portions have not received significant attention due to their high complexity.*
- *Optimization and simulation make the design cycle longer.*
- *This paper presents a new approach to reduced design optimization time with use of metamodels.*
- *Metamodels allow fast design space exploration and reduce the design cycle time.*
- *Three different optimization algorithms are compared: Exhaustive search, Tabu search, and Simulated Annealing. The algorithms are analysed to determine their stability for metamodeling-based optimization and are compared to simulation based optimization approaches.*
- *It is observed that the metamodel based simulated annealing optimization achieved ~9000x speed-up over the actual circuit based optimization.*

Introduction

A metamodel is a mathematical formula that represents the circuit's behavior within a given range of parameters and is derived from sampling points.

The metamodel considered has the form:

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

Where y is the response being modeled (e.g. frequency), x = [W_n,W_p] is the vector of variables and a_{ij} are the coefficients. The multinomial regression is determined for k=4.

A metamodel is created on a full RCLK (resistance, capacitance, self and mutual inductance) parasitic extracted netlist, for each figure of merit.

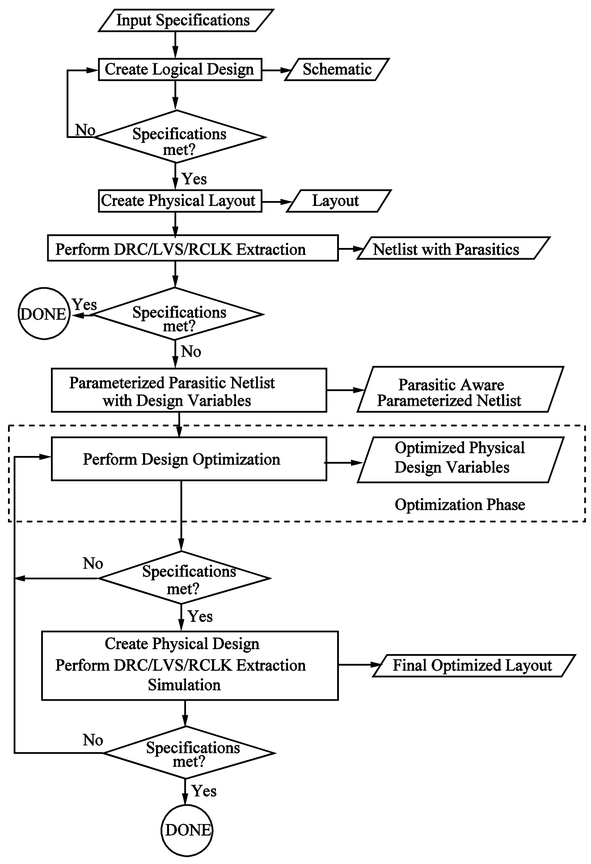
This study is attempting to answer two main questions for mixed signal circuits:

•How fast can design space exploration be performed?

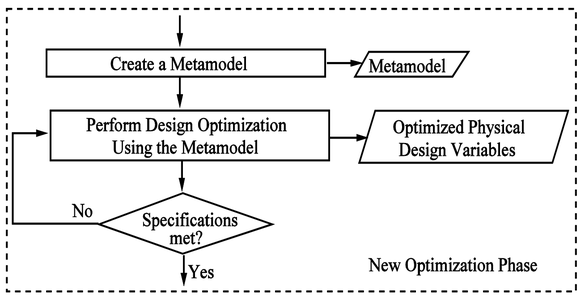
•How fast can layout generation and optimization be performed?

The Proposed Methodology for Metamodeling Optimization

The flowchart presents our parasitic aware accurate physical design optimization flow.

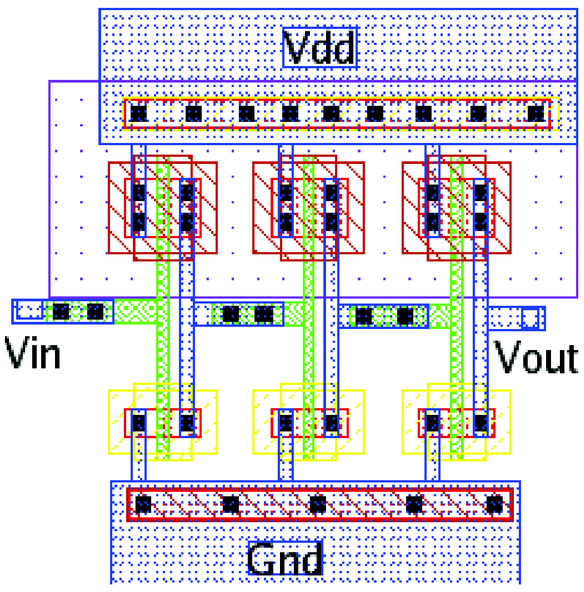


The optimization flow above is altered to include the metamodel. The section with dashed line is replaced by:



Case Study Circuit

The proposed approach has been used on a ring oscillator (RO) designed in 45 nm technology. The figure below shows the physical layout of the test circuit.



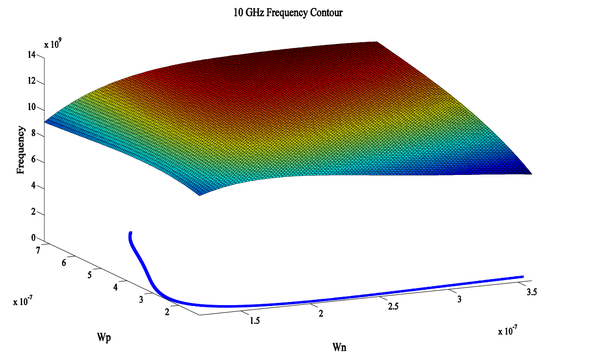
The RO has been characterized for frequency, power and power frequency ratio (PRF) with width of PMOS and NMOS (*Wn* and *Wp*) as design variables.

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

Exhaustive Search

Algorithm 1 Exhaustive Search Algorithm for W_n and W_p

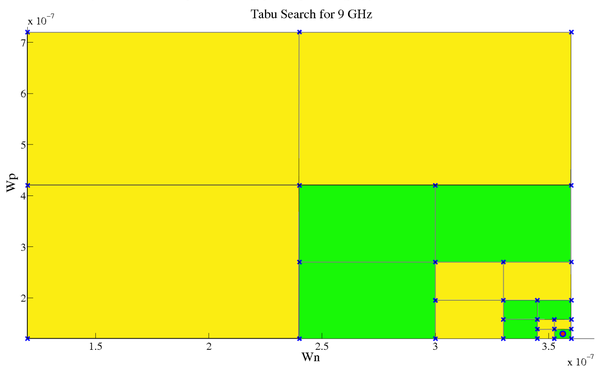
1: Determine the step size $Step$ needed for each variable between W_{nmax}, W_{nmin} and W_{pmax}, W_{pmin} for N amount of simulations
2: initialize the result counter $result\_counter = 0$
3: **for** ($i = W_{nmin}$ to W_{nmax} with $Step_{Wn}$) **do**
4: **for** ($j = W_{pmin}$ to W_{pmax} with $Step_{Wp}$) **do**
5: $S_{ij} = F(i, j)$
6: calculate and record minimum (optional)
7: calculate and record maximum (optional)
8: calculate PFR (optional)
9: **if** (value is within the limit) **then**
10: $result[result\_counter] = S_{ij}$
11: $result\_counter = resut\_counter + 1$
12: **end if**
13: **end for**
14: **end for**
15: Return $result$, minimum, maximum and PFR (optional)



Tabu Search

Algorithm 2 Tabu Search Algorithm for W_n and W_p

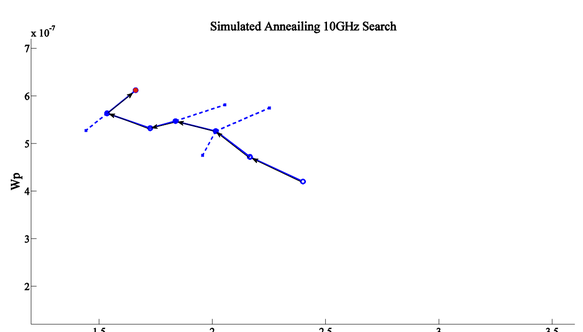
1: Initialize iteration counter Counter = 0
2: Conduct DOE analysis for W_n and W_p
3: Generate initial feasible solution S_i
4: **while** ($Counter$ <Max_Counter) **do**
5: Generate the next feasible solution S_i^*
6: $Counter = Counter + 1$
7: **if** (S_i is not visited in the previous iterations) **then**
8: **if** (S_i^* is better solution than S_i) **then**
9: **if** (result is found) **then**
10: break while loop
11: **end if**
12: $S_i = S_i^*$
13: **else**
14: Discard the Solution S_i^*
15: **end if**
16: **end if**
17: **end while**



Simulated Annealing

Algorithm 3 Simulated Annealing Algorithm for W_n and W_p

1: Initialize iteration counter Counter = 0
2: Initialize first feasible solution S_i=F(mid(W_n),mid(W_p))
3: Determine initial $Cost_i$ for the solution S_i
4: Initialize temperature T as T_i
5: **while** ($Cost$ is varying) **do**
6: $Counter$ =Maximum number of iterations
7: **while** ($Counter > 0$) **do**
8: Generate random transition from S_i to S_i^*
9: **if** (S_i^* is acceptable solution) **then**
10: $result = S_i^*$
11: break both while loops
12: **else**
13: Calculate change in cost as: $\Delta_{Cost} = Cost_S - Cost_i^*$
14: **if** ($\Delta_{Cost} < 0$ random(0,1) $< e^{\frac{\Delta Cost}{T}}$) **then**
15: Update the solution with new solution, $S \leftarrow S_i^*$
16: **end if**
17: **end if**
18: $Counter = Counter - 1$
19: **end while**
20: Decrease temperature as: $T = T*$Cooling_Rate
21: **end while**
22: **return** $result$



Comparison Results

TABLE II
EXHAUSTIVE SEARCH OPTIMIZATION FOR FREQUENCY OF 10 GHZ WITH 1% ACCURACY

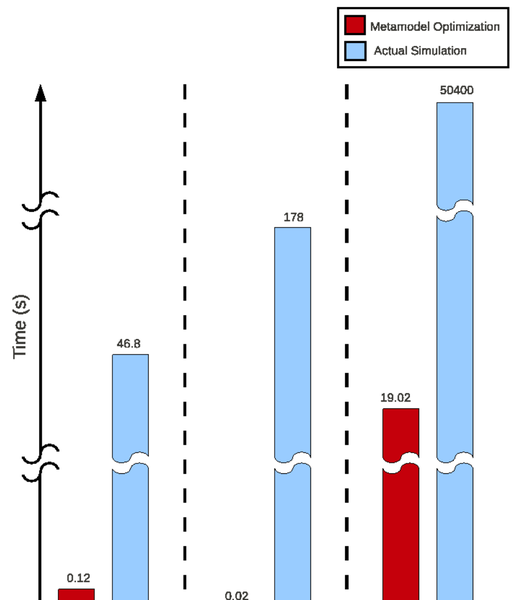
Iterations	Points Found	Times	Min Power	Max Frequency	Min PFR
Parasitic Netlist Optimization					
10000	42	32 hours	19.9μW	12.7 GHz	2.18e-15
2500	13	8 hours	19.9μW	12.7GHz	2.18e-15
625	2	2 hours	19.9μW	12.7GHz	2.18e-15
Metamodel Optimization					
1000000	4566	57.01 sec	19.9 μW	12.8 GHz	2.18e-15
250000	1142	21.73 sec	19.9μW	12.8 GHz	2.18e-15
10000	44	0.46 sec	19.9μW	12.7 GHz	2.18e-15
2500	13	0.04 sec	19.9μW	12.7GHz	2.18e-15
625	2	0.02 sec	19.9μW	12.7GHz	2.18e-15

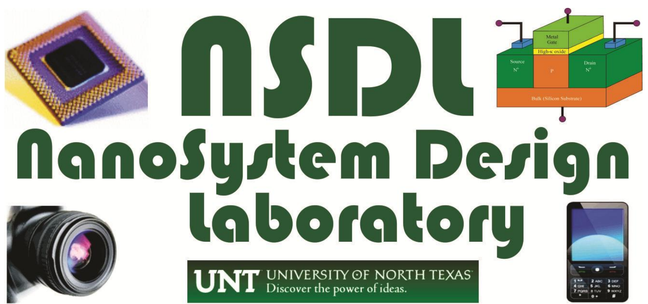
TABLE III
TABU SEARCH OPTIMIZATION FOR FREQUENCY

Number of Simulations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization				
32	9 GHz	9.38 GHz	4.22%	6.25 min
7	9.5 GHz	9.4 GHz	1.05%	1.37 min
12	10 GHz	9.94 GHz	0.62%	2.34 min
18	10.5 GHz	10.5 GHz	0.32%	3.52 min
10	11 GHz	11.1 GHz	0.84%	1.95 min
19	11.5 GHz	11.4 GHz	0.71%	3.71 min
30	12 GHz	11.8 GHz	1.92%	5.86 min
4	12.5 GHz	12.6 GHz	0.96%	0.78 min
Metamodeling Optimization				
30	9 GHz	9.4 GHz	4.41%	8.6 ms
7	9.5 GHz	9.41 GHz	0.94%	6.05 ms
12	10 GHz	9.93 GHz	0.74%	7.18 ms
24	10.5 GHz	10.5 GHz	0.32%	7.38 ms
10	11 GHz	11.1 GHz	0.84%	6.41 ms
19	11.5 GHz	11.4 GHz	0.71%	7.11 ms
30	12 GHz	11.8 GHz	1.92%	9.3 ms

TABLE IV
SIMULATED ANNEALING OPTIMIZATION FOR FREQUENCY

Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization				
35	9 GHz	8.97 GHz	0.33%	6.84 min
14	9.5 GHz	9.44 GHz	0.63%	2.73 min
15	10 GHz	10.07 GHz	0.31%	2.93 min
24	10.5 GHz	10.40 GHz	0.97%	4.69 min
16	11 GHz	10.96 GHz	0.36%	3.12 min
5	11.5 GHz	11.46 GHz	0.34%	0.98 min
3	12 GHz	11.99 GHz	0.08%	0.59 min
10	12.5 GHz	12.47 GHz	0.24%	1.95 min
Metamodeling Optimization				
32	9 GHz	8.96 GHz	0.48%	1.8 ms
18	9.5 GHz	9.41 GHz	0.94%	1.05 ms
10	10 GHz	10.05 GHz	0.48%	0.77 ms
19	10.5 GHz	10.40 GHz	0.96%	1.16 ms
13	11 GHz	10.95 GHz	0.49%	0.85 ms
4	11.5 GHz	11.48 GHz	0.22%	0.38 ms
2	12 GHz	11.98 GHz	0.16%	0.16 ms
12	12.5 GHz	12.42 GHz	0.63%	0.95 ms





UNT
UNIVERSITY OF NORTH TEXAS
Discover the power of ideas