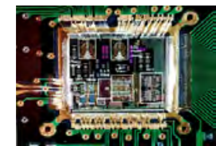
Lecture 9: Ultra-Fast Design of Ring Oscillator

CSCE 6933/5933
Advanced Topics in VLSI Systems

Instructor: Saraju P. Mohanty, Ph. D.

NOTE: The figures, text etc included in slides are borrowed from various books, websites, authors pages, and other sources for academic purpose only. The instructor does not claim any originality.



Outline of the talk

- Introduction
- Metamodeling
- Fast Design Exploration through Metamodeling
- Metamodeling based Design Flow
- Sampling Techniques
- Ring Oscillator Example Circuit
- "Golden" Surface
- Experimental Results
- Conclusions and Future Research

Introduction

- Complex computations for analog circuits to include parasitics
- Physical layout and simulation analysis is very costly processes in design flow
- Metamodeling is mathematical formula that represents circuit's behavior within a given range using sampling points
- This paper targets sampling techniques which are technology independent and the amount that is needed to create an accurate metamodel

The Metamodels

- Mathematical representation of output.
- Prediction equation:

$$(\hat{F}(x_n) \approx F(x_n)) = F(x_n) + \varepsilon$$

- Can be used in different tool like MATLAB.
- Metamodels are not as computationally expensive than simulations.

The 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.
- Metamodel has the form:

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

- Where y is the response being modeled (e.g. frequency), $x = [W_n, W_p]$ is the vector of variables and α_{ij} are the coefficients.
- The polynomial regression determined k = 2 for DOE and k = 4 for other cases.

Error Analysis

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y(x_k) - \hat{y}(x_k)\right)^2},$$

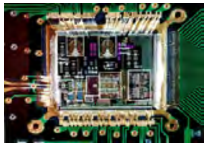
$$\sigma = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\left|y(x_k) - \hat{y}(x_k)\right| - RMSE\right)^2},$$

Where:

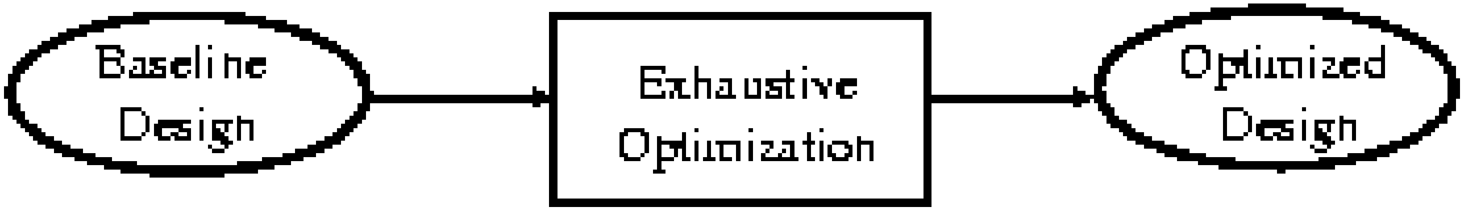
x_k is the set of parameters

$\hat{y}$ is the predicted equation

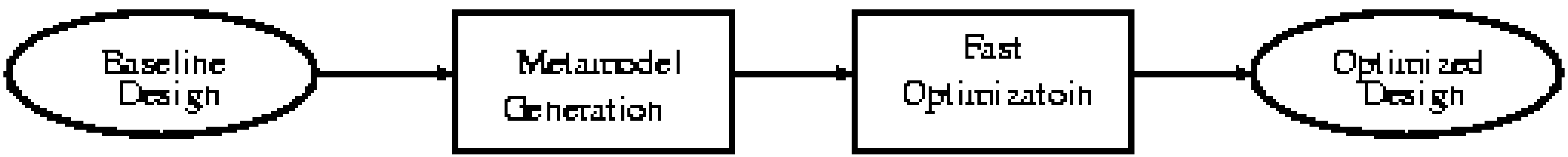
N is the number of sampling points



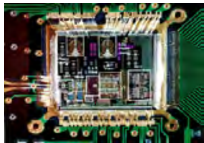
Fast Design Exploration Through Metamodeling



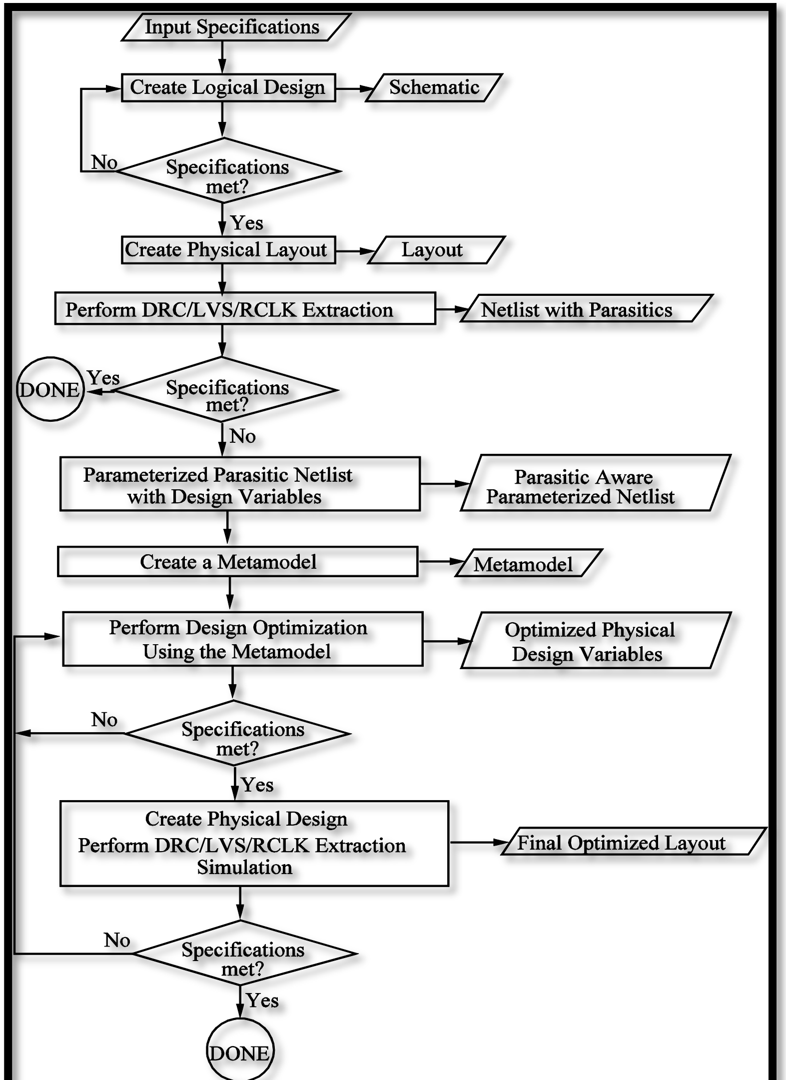
a. Traditional Slow Approach



b. Metamodeling-Based Fast Approach



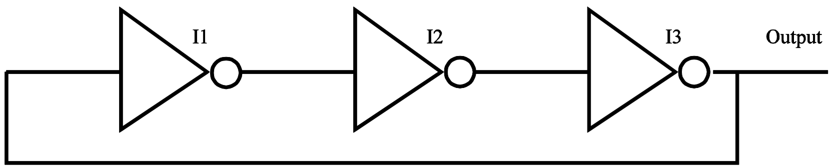
Metamodeling Design Flow

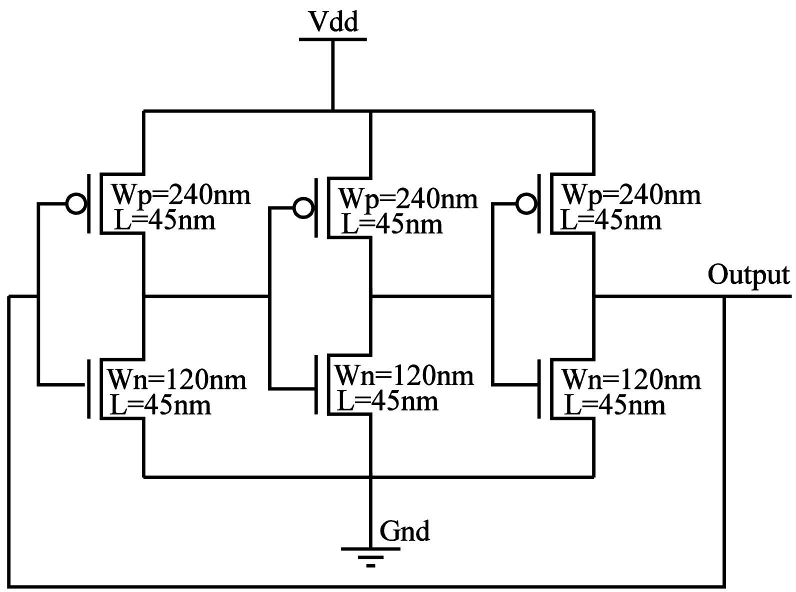


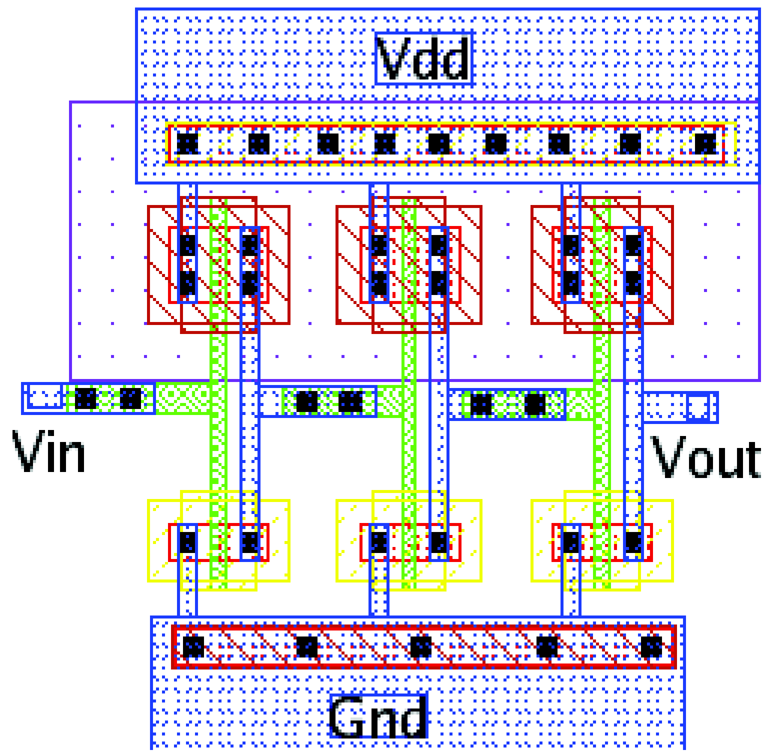
- Regular design flow is altered for using metamodels.
- Advantages for using metamodels:
  - Reusability
  - Speed
  - Accuracy
- Physical design is done only 2 times in the proposed design flow.

Ring Oscillator: 45nm CMOS Design

$$f = \frac{1}{2Nt_p}$$ Where f - frequency of oscillations, N - number of inverters, t_p - delay of each inverter







Ring Oscillator: Characterization

Eye Diagram for Parasitic Aware Netlist Simulation

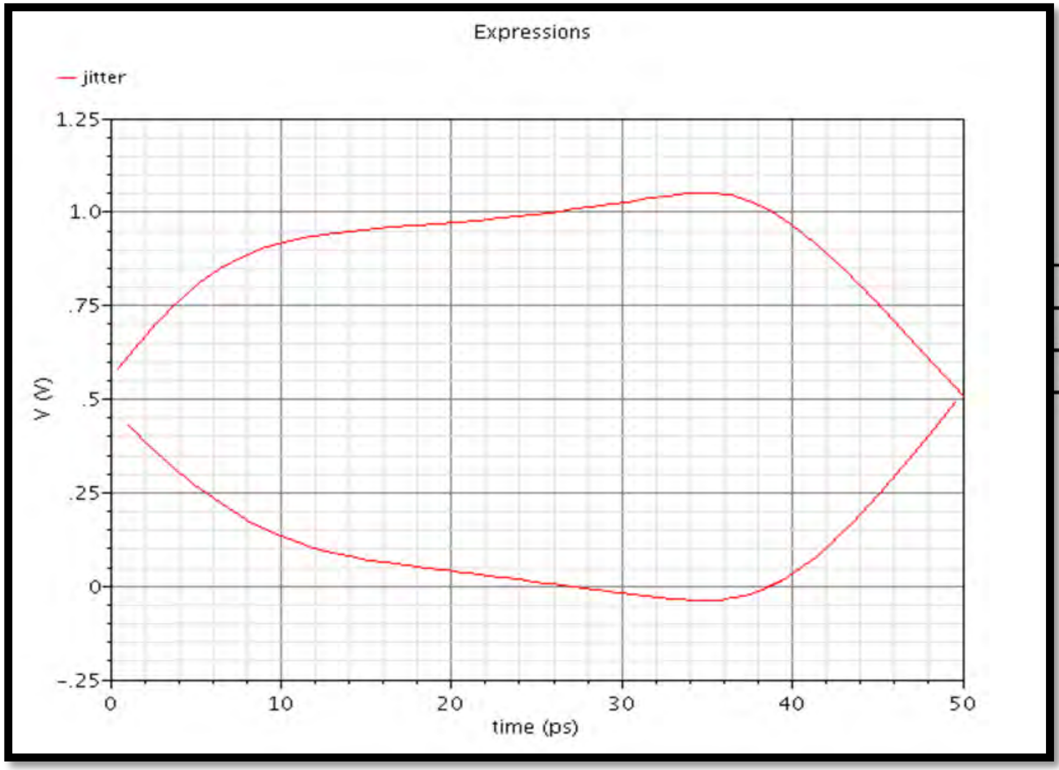


TABLE II
SIMULATION COMPARISON

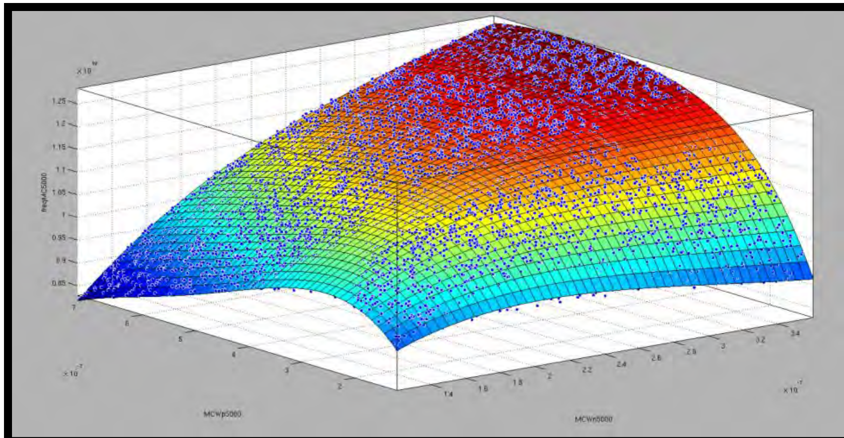
Extraction	Power	Frequency
Schematic	27.17 μW	16.21 GHz
120nm-240nm Parasitic	26.96 μW	9.88 GHz

Sampling Techniques Explored

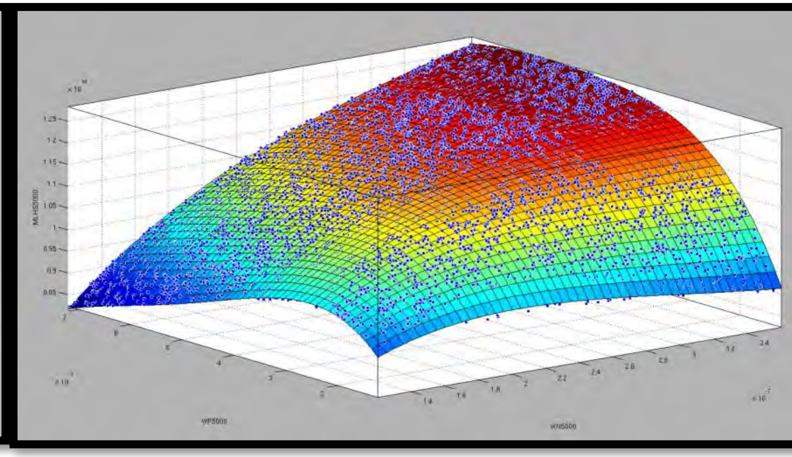
- Exhaustive – evenly distributed large amount of samples
- Monte Carlo – random sampling
- Latin Hypercube – random sampling within each Latin square
- Middle Latin Hypercube – middle point sampling within each Latin square
- Design of Experiments – min, mid, max sampling for each parameter

Sampling Techniques: Applied to Ring Oscillator Circuit

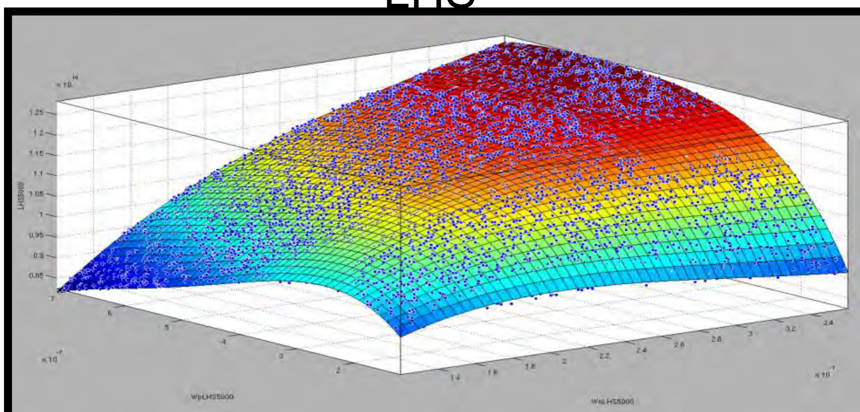
Monte Carlo



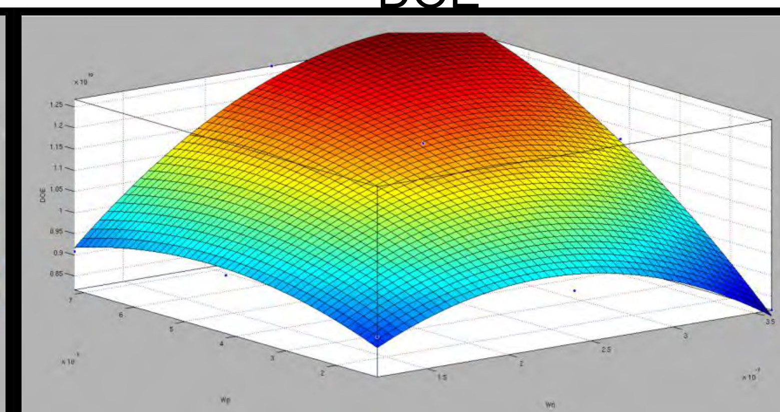
MLHS



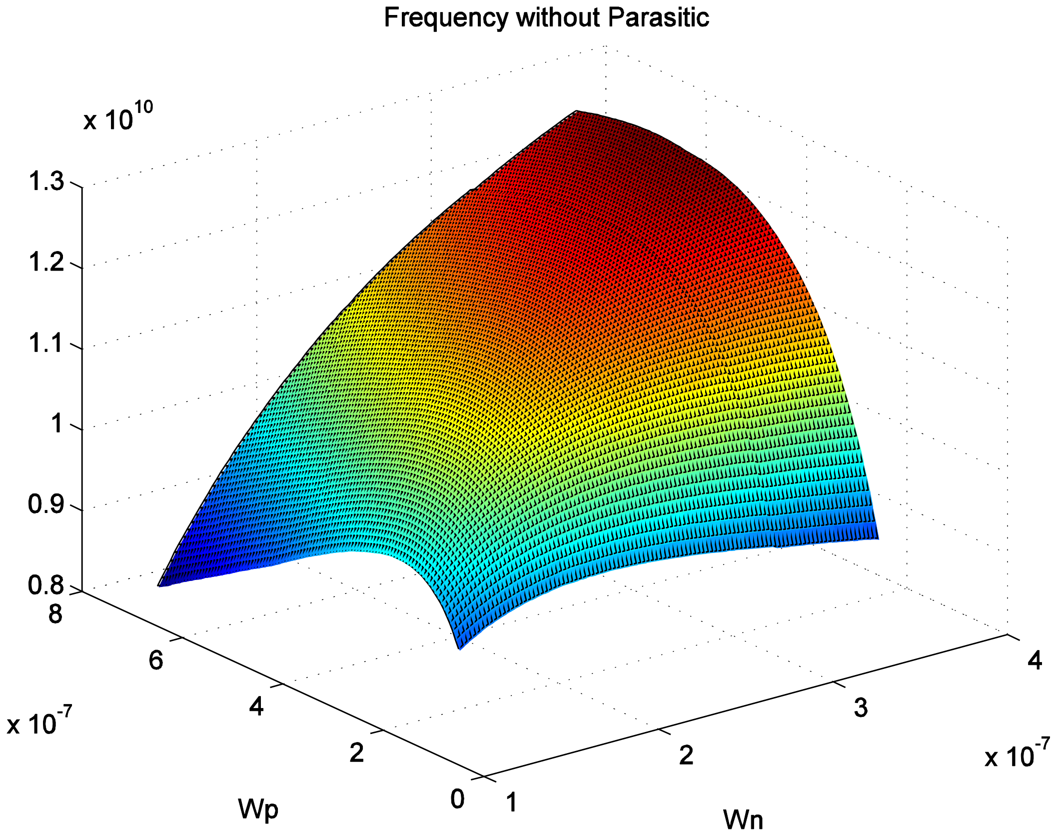
LHS



DOE



"Golden" Surface

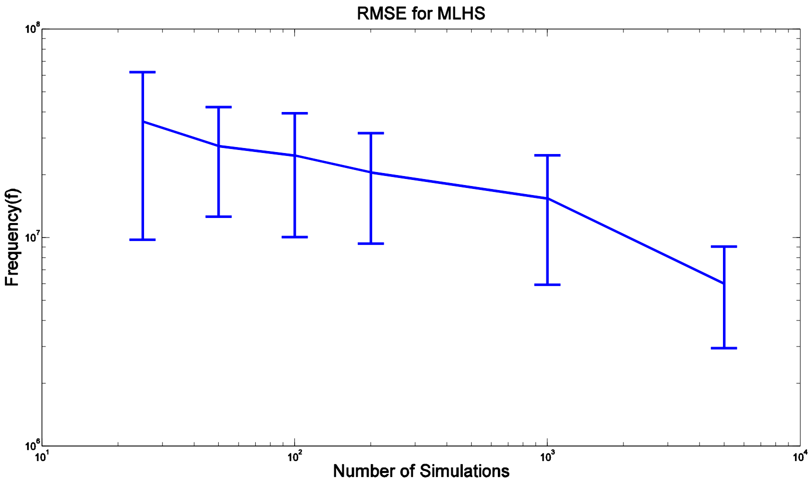


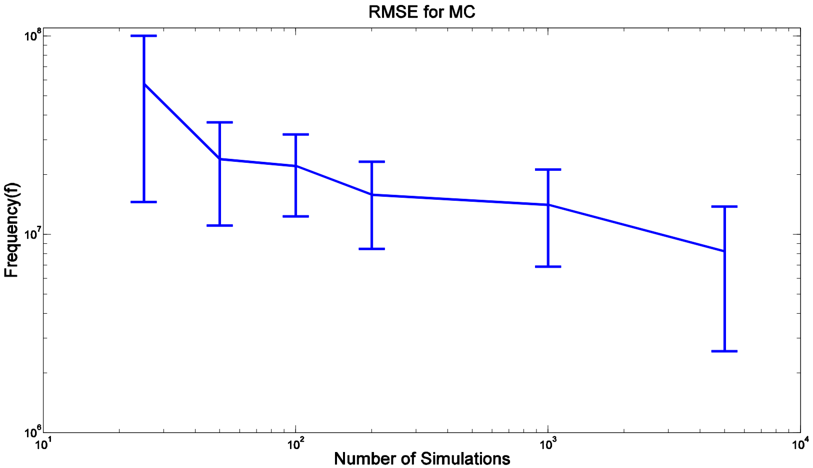
10,000 sampling surface was created for exhaustive analysis.

-RMSE < 0.01%

"Golden" Surface is used as actual results to compare to the sampling techniques.

Experimental Results





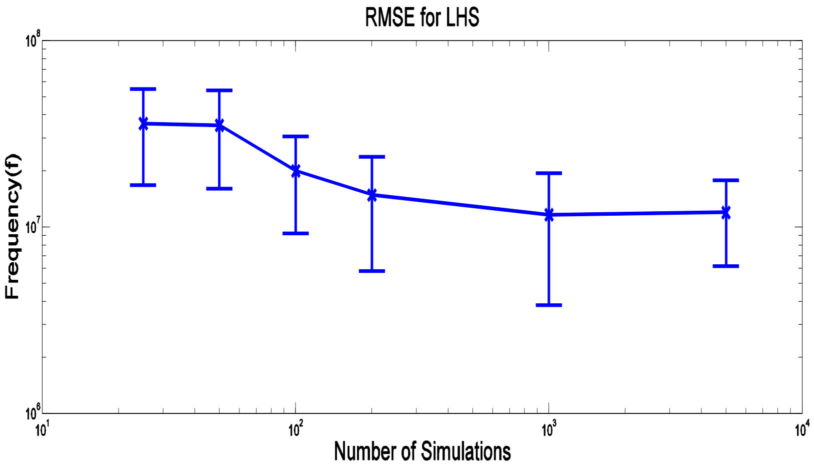
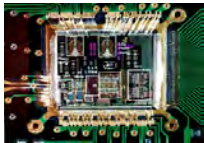


TABLE III
RMSE COMPARISON FOR DIFFERENT SAMPLING TECHNIQUES (IN MHZ)

Samples	MC		LHS		MLHS	
N	μ	σ	μ	σ	μ	σ
25	57.5	42.9	35.6	19.1	36.0	26.2
50	24.0	12.9	35.2	19.1	27.4	14.8
100	22.1	9.79	20.0	10.7	24.8	14.7
200	15.9	7.39	14.9	9.04	20.5	11.2
1000	14.1	7.21	11.7	7.81	15.4	9.44
5000	8.20	5.62	12.0	5.84	5.99	3.04

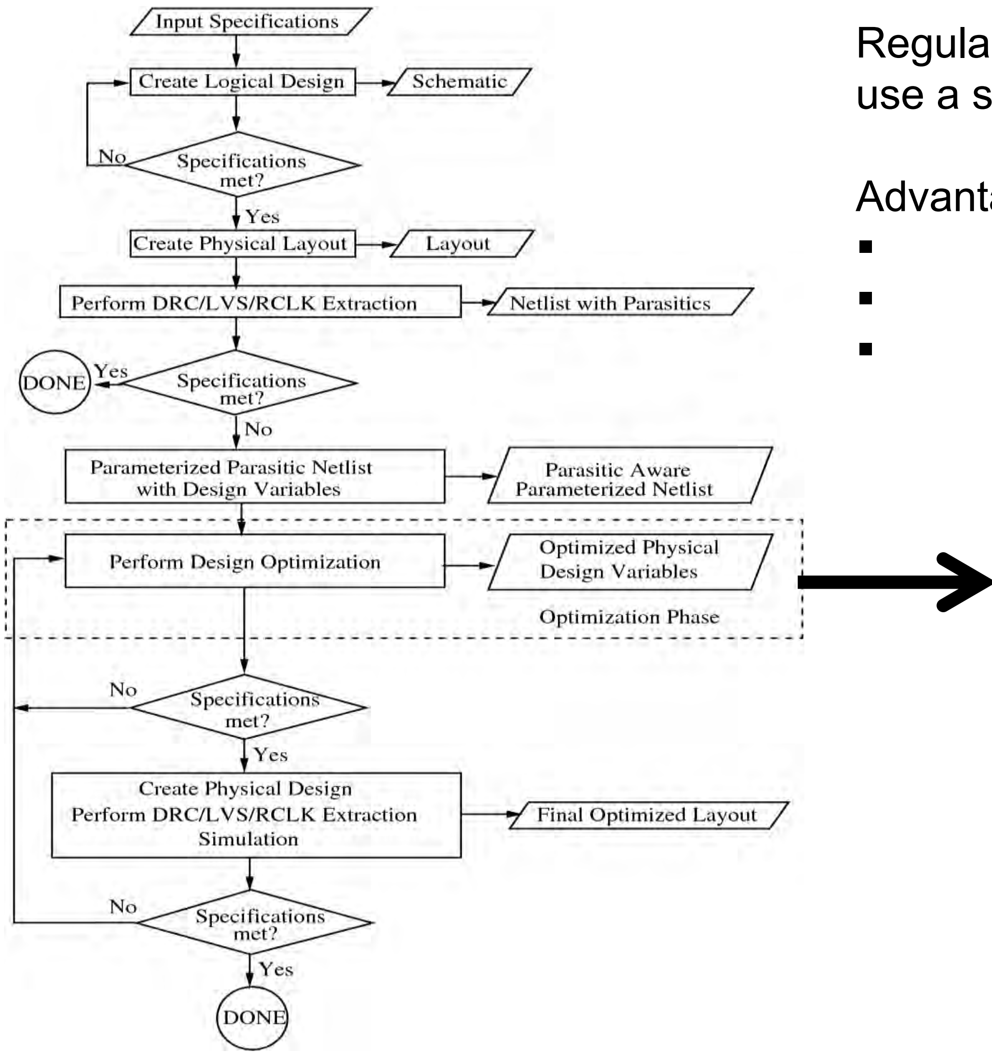


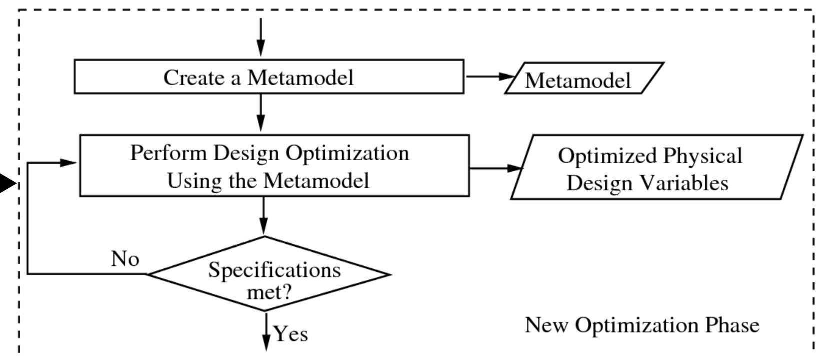
Metamodeling Optimization

Regular optimization design flow is altered to use a surrogate model (metamodel).

Advantages for using metamodels:

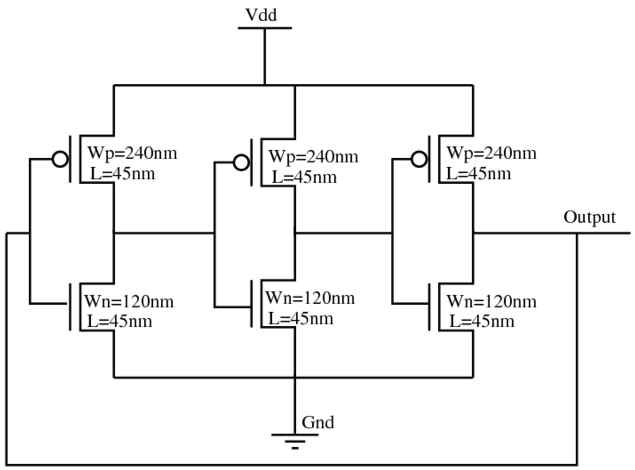
- Reusability
- Speed
- Accuracy



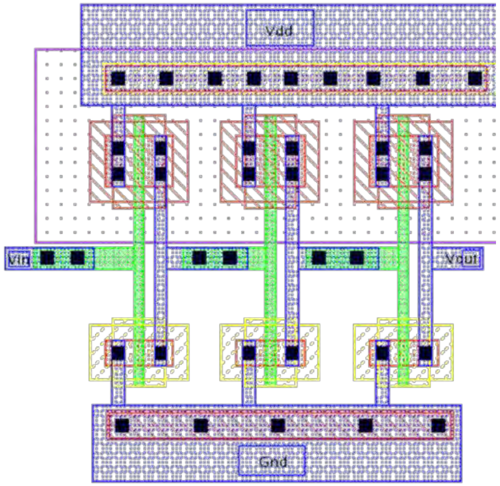


Metamodeling Optimization

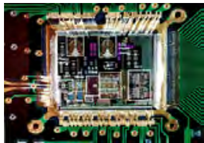
- Implementation using 45 nm Ring Oscillator including schematic and physical layout as a test circuit.
- Multiple sampling techniques are compared (LHS, MLHS, Exhaustive Sampling, Monte Carlo)
- Optimization techniques are compared (Exhaustive Search, Simulated Annealing, Tabu Search)



Schematic of Ring Oscillator



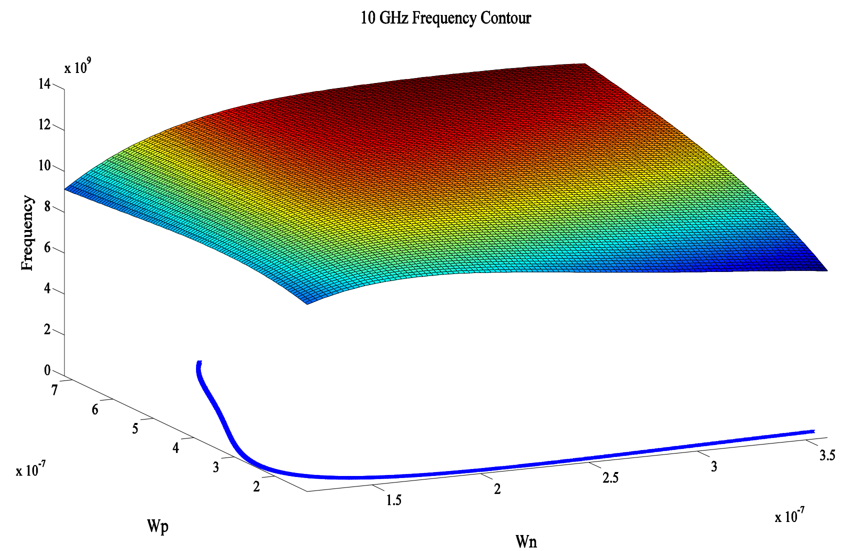
Physical Layout of Ring Oscillator



Exhaustive Search

Algorithm 1 Exhaustive Search Algorithm for W_n and W_p

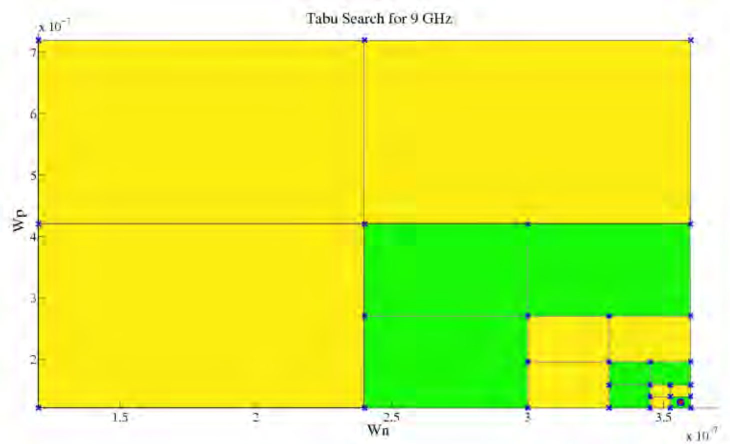
```
Determine the step size Step needed for each variable between W_nmax, W_nmin and W_pmax, W_pmin for N amount of simulations
initialize the result counter result_counter = 0
for (i = W_nmin to W_nmax with Step_Wn) do
    for (j = W_pmin to W_pmax with Step_Wp) do
        S_ij = F(i, j)
        calculate and record minimum (optional)
        calculate and record maximum (optional)
        calculate PFR (optional)
        if (value is within the limit) then
            result[result_counter] = S_ij
            result_counter = resut_counter + 1
        end if
    end for
end for
Return result, minimum, maximum and PFR (optional)
```



Tabu Search

Algorithm 2 Tabu Search Algorithm for W_n and W_p

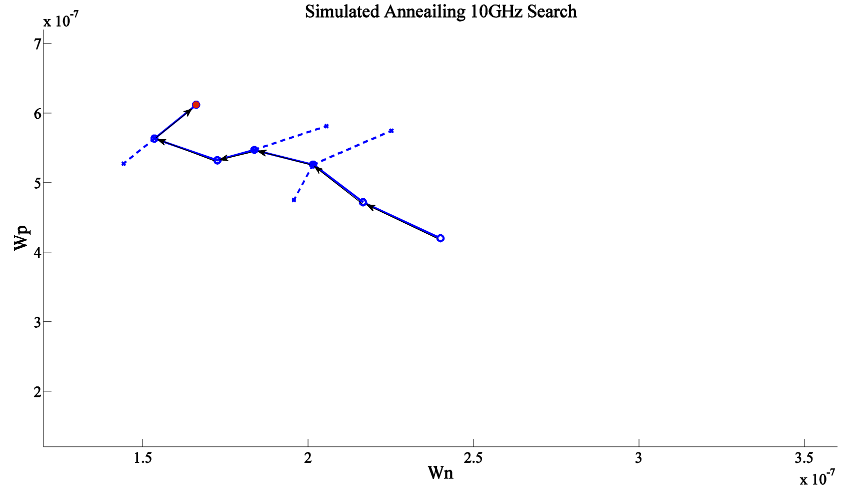
```
Initialize iteration counter Counter = 0
Conduct DOE analysis for W_n and W_p
Generate initial feasible solution S_i
while (Counter <Max_Counter) do
    Generate the next feasible solution S_i^*
    Counter = Counter + 1
    if (S_i is not visited in the previous iterations) then
        if (S_i^* is better solution than S_i) then
            if (result is found) then
                break while loop
            end if
            S_i = S_i^*
        else
            Discard the Solution S_i^*
        end if
    end if
end while
return  result or mid point of S_i
```

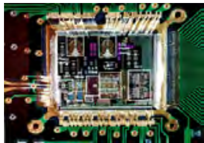


Simulated Annealing

Algorithm 3 Simulated Annealing Algorithm for W_n and W_p

Initialize iteration counter Counter = 0
Initialize first feasible solution S_i=F(mid(W_n),mid(W_p))
Determine initial $Cost_i$ for the solution S_i
Initialize temperature T as T_i
while ($Cost$ is varying) **do**
  $Counter$ =Maximum number of iterations
  while ($Counter > 0$) **do**
    Generate random transition from S_i to S_i^*
    if (S_i^* is acceptable solution) **then**
      $result = S_i^*$
      break both while loops
    else
      Calculate change in cost as: $\Delta_{Cost} = Cost_S - Cost_i^*$
      if ($\Delta_{Cost} < 0$ random(0,1) $< e^{\frac{\Delta_{Cost}}{T}}$) **then**
        Update the solution with new solution, $S \leftarrow S_i^*$
      end if
    end if
    $Counter = Counter - 1$
  end while
  Decrease temperature as: $T = T *$Cooling_Rate
end while
return $result$





Metamodeling Optimization Results

- Optimization techniques using parasitic netlist and metamodeling are compared.

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

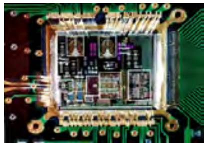
Iterations	Points Found	Times	Min Power	Max Frequency	Min PFR
Parasitic Netlist Optimization					
10000	42	32 hours	19.9μW	12.7 GHz	2.18e-15
2500	13	8 hours	19.9μW	12.7GHz	2.18e-15
625	2	2 hours	19.9μW	12.7GHz	2.18e-15
Metamodel Optimization					
1000000	4566	57.01 sec	19.9 μW	12.8 GHz	2.18e-15
250000	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 IV
SIMULATED ANNEALING OPTIMIZATION FOR FREQUENCY

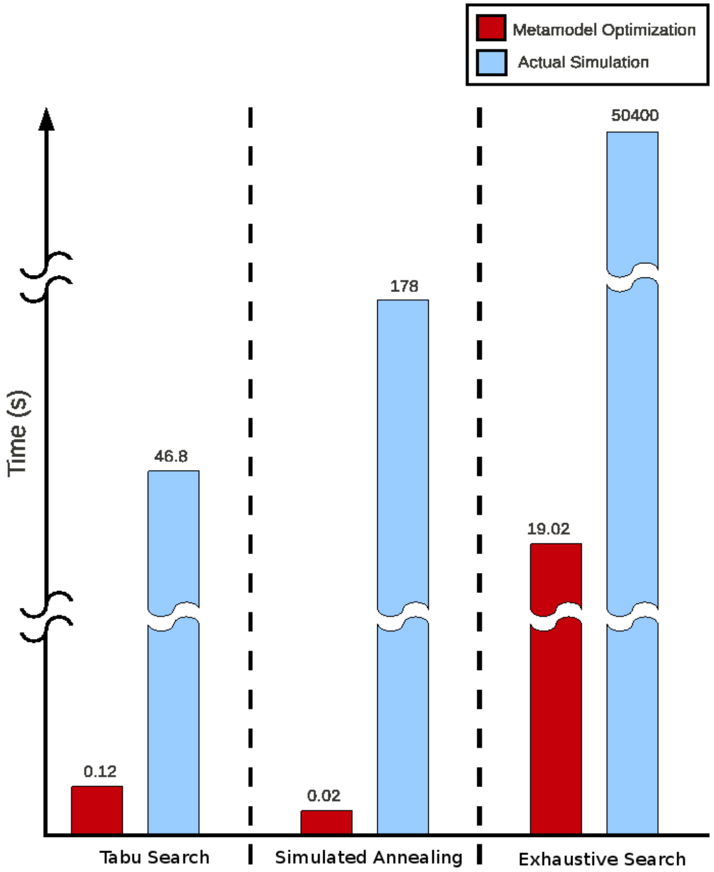
Loop Iterations	Results Needed	Results Found	Accuracy	Time
Parasitic Netlist Optimization				
35	9 GHz	8.97 GHz	0.33%	6.84 min
14	9.5 GHz	9.44 GHz	0.63%	2.73 min
15	10 GHz	10.07 GHz	0.31%	2.93 min
24	10.5 GHz	10.40 GHz	0.97%	4.69 min
16	11 GHz	10.96 GHz	0.36%	3.12 min
5	11.5 GHz	11.46 GHz	0.34%	0.98 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

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



Comparison of the running time of the three algorithms



- **Optimization without metamodels:** the tabu-search optimization is faster by 1077× than the exhaustive search and 3.8× faster than the simulated annealing optimization.
- **Optimization with metamodels:** the simulated annealing optimization is faster by 951× than the exhaustive search and 6× faster than the tabu search optimization.

Conclusions

- A design flow for metamodeling is proposed.
- A 45nm ring oscillator was subjected to the proposed design flow.
- Uniform sampling techniques has better performance than DoE or than randomized.
- Designers should choose LHS or MLHS over MC but the trend in typical design environments is the opposite.