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

G. Zheng[1], S. P. Mohanty[2], E. Kougianos[3], and O. Garitselov[4]
NanoSystem Design Laboratory (NSDL, http://nsdl.cse.unt.edu)
University of North Texas, Denton, TX 76203, USA.[1,2,3,4]

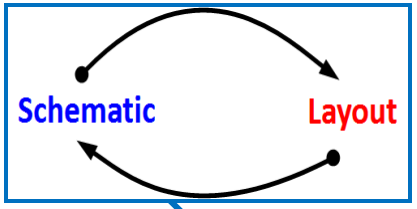
Presenter:
Geng Zheng
University of North Texas
Email: gengzheng@my.unt.edu

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

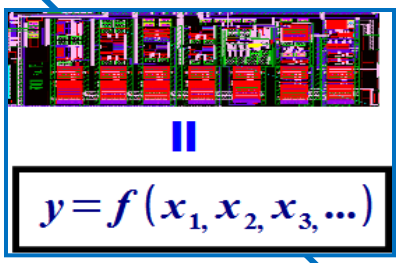
NSDL NanoSystem Design laboratory UNT UNIVERSITY OF NORTH TEXAS Discover the power of ideas

UNT UNIVERSITY OF NORTH TEXAS Discover the power of ideas

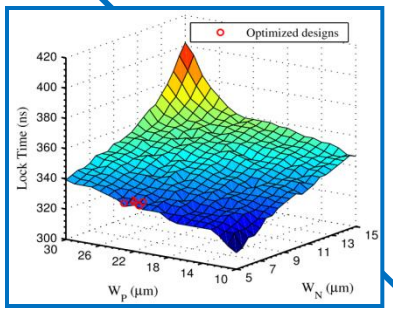
Agenda



Motivation



Metamodeling



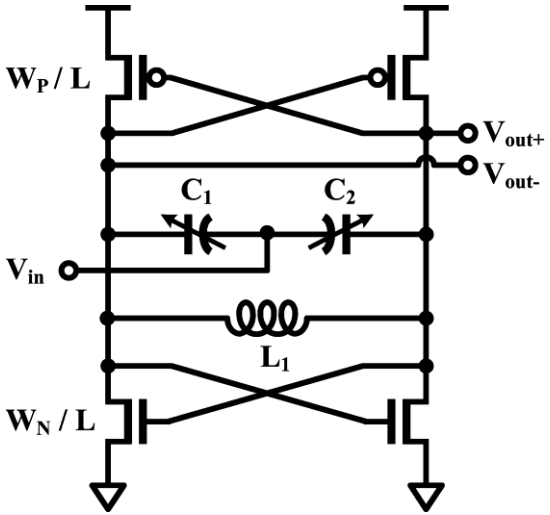
Case study: PLL Optimization

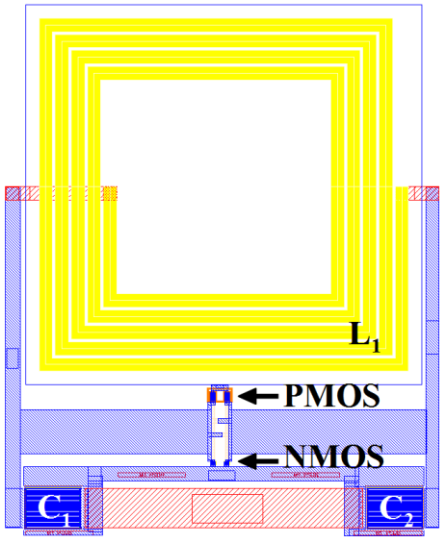
Conclusions

Motivation

- **Parasitics greatly impact nano-CMOS circuit designs**

Schematic





Layout

LC VCO

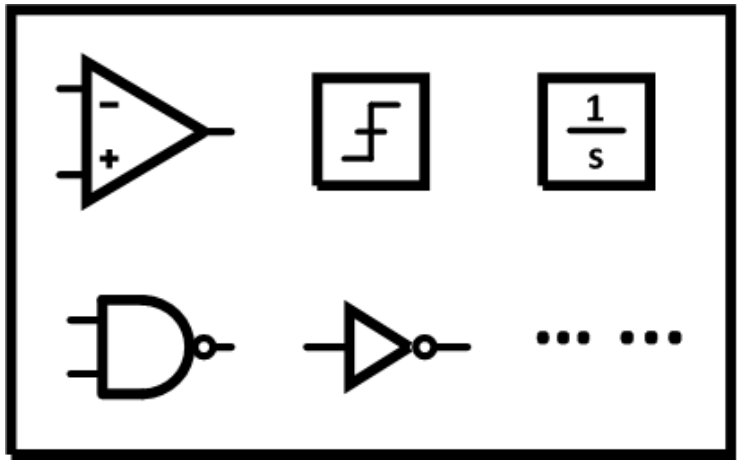
180 nm CMOS

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

Mixed-signal Circuit Simulation

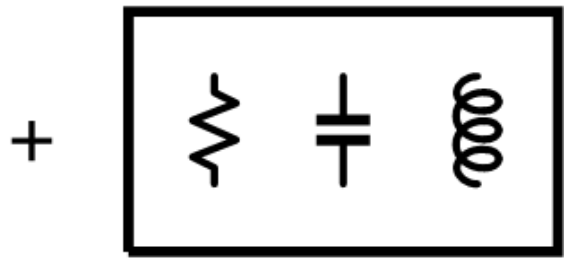
- **Mixed-signal layout simulations require great amounts of computation time and effort**

Mixed-signal Systems



Linear and non-linear blocks

Parasitics

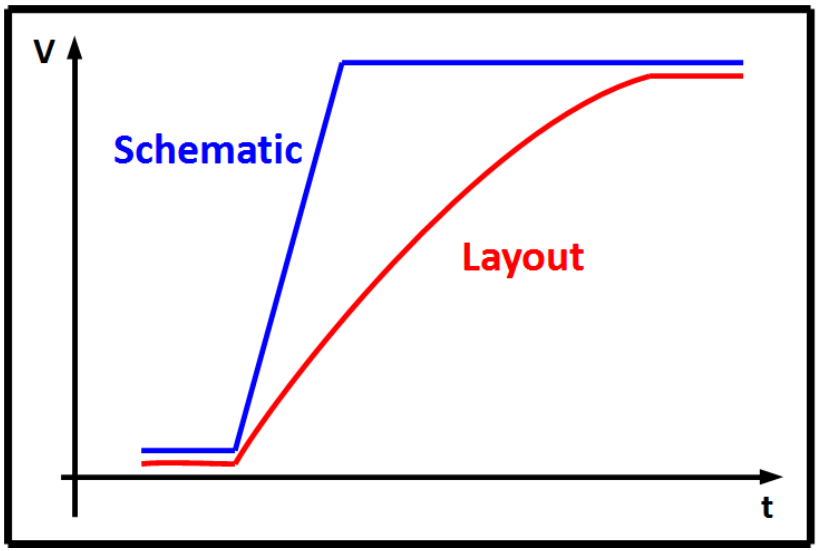


Lots of parasitic components

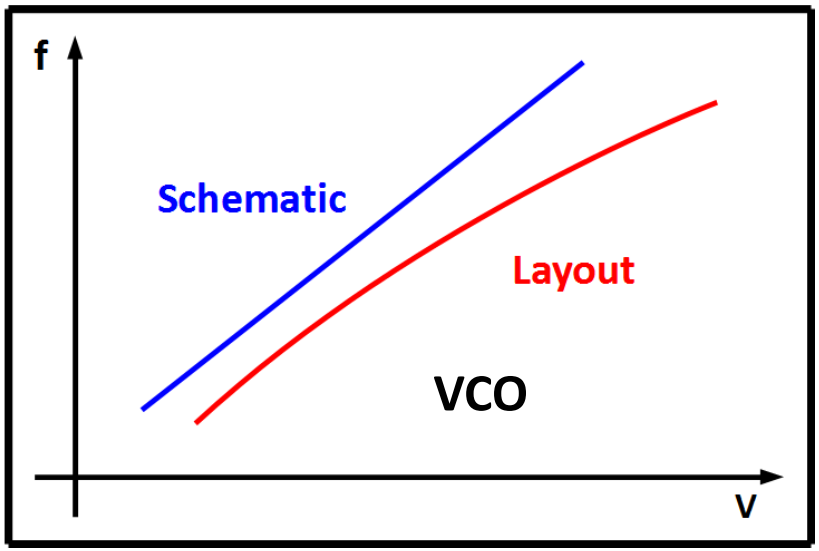
Effects of Parasitics

- **Parasitics cause significant mismatch between schematic and layout simulations**

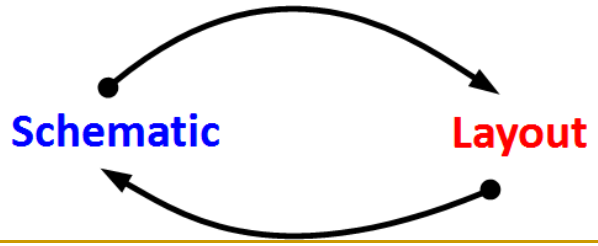
Digital Circuit



Analog Circuit



Numerous design iterations are required until specifications are met

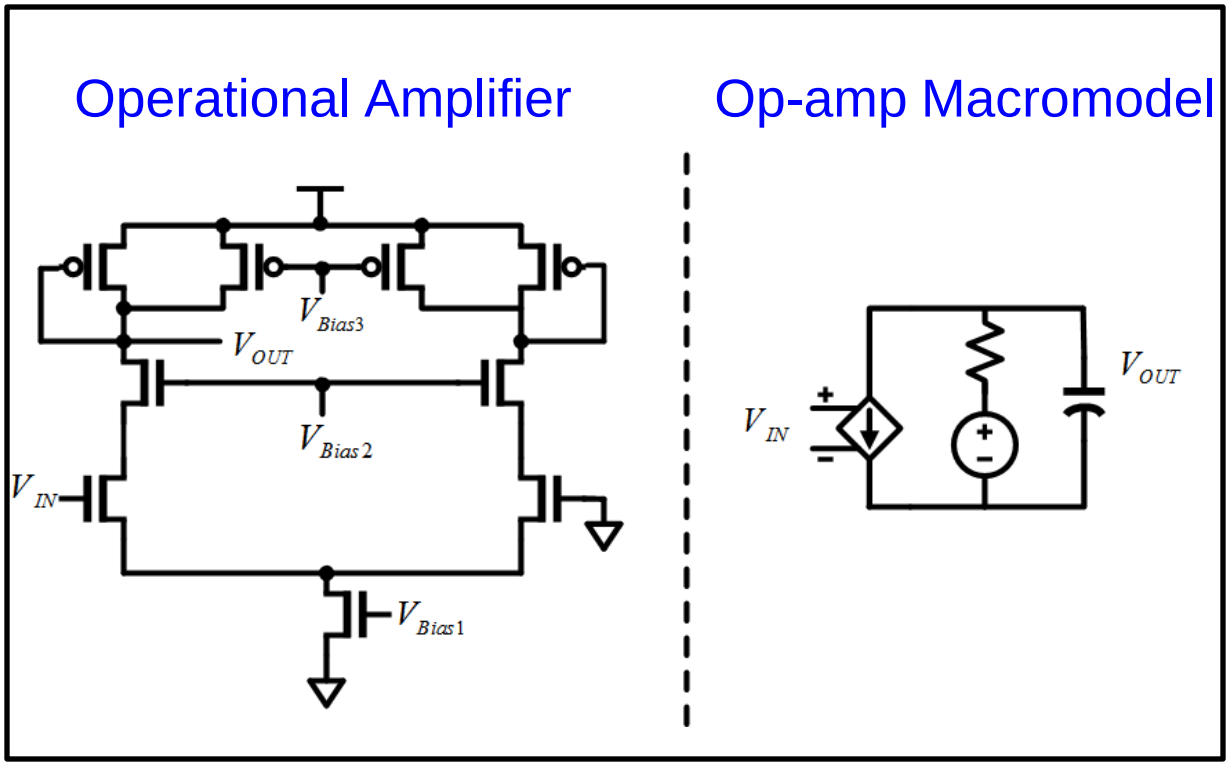


Each iteration requires layout modification and re-simulation

Behavioral Model

- **Marcomodeling based behavioral models can reduce simulation time but have several disadvantages**

Macromodeling



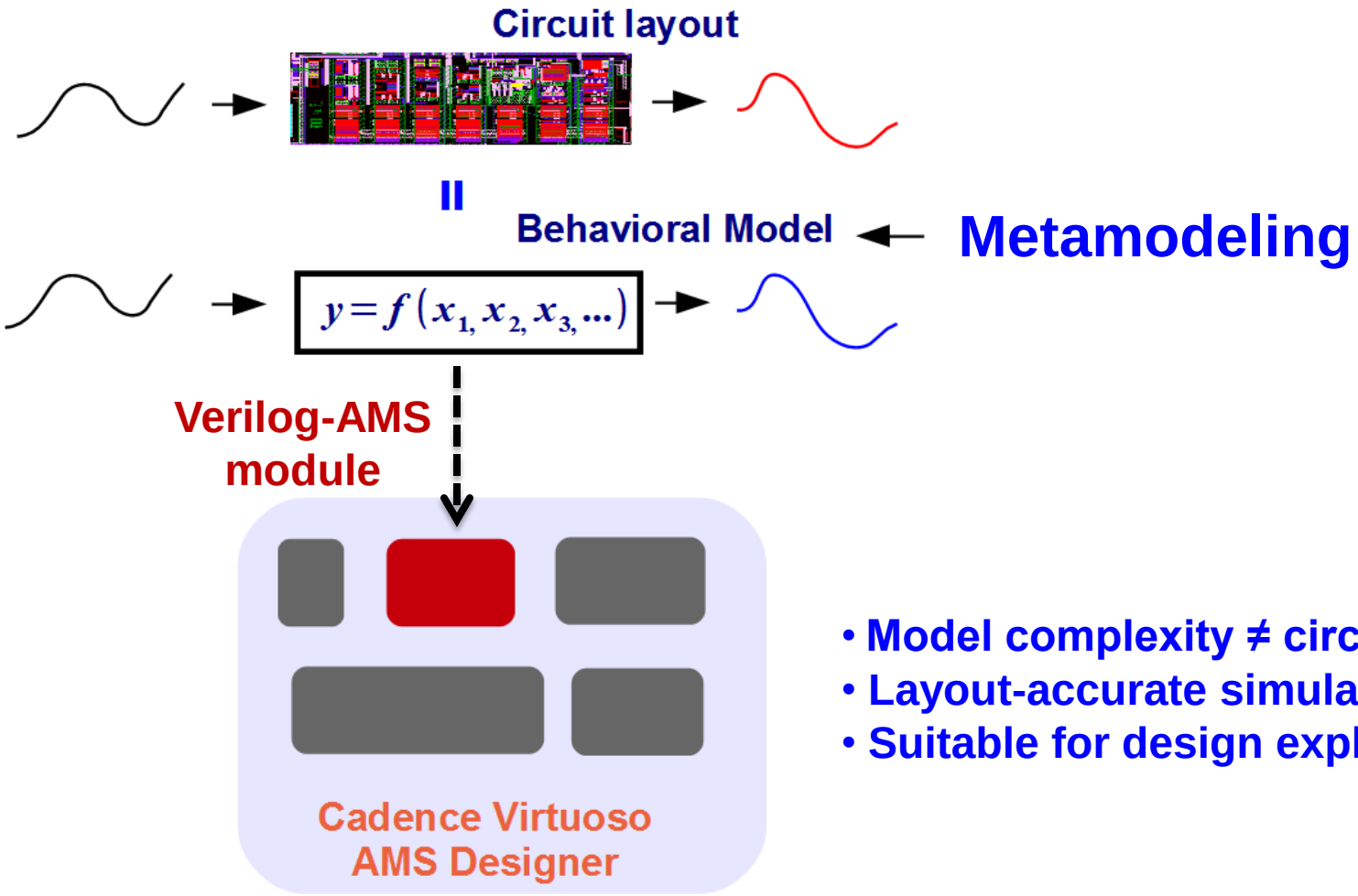
- **complexity limitation**

Not suitable for

- **paracitics incorporation**
- **design exploration**

Parasitic-Aware Metamodel

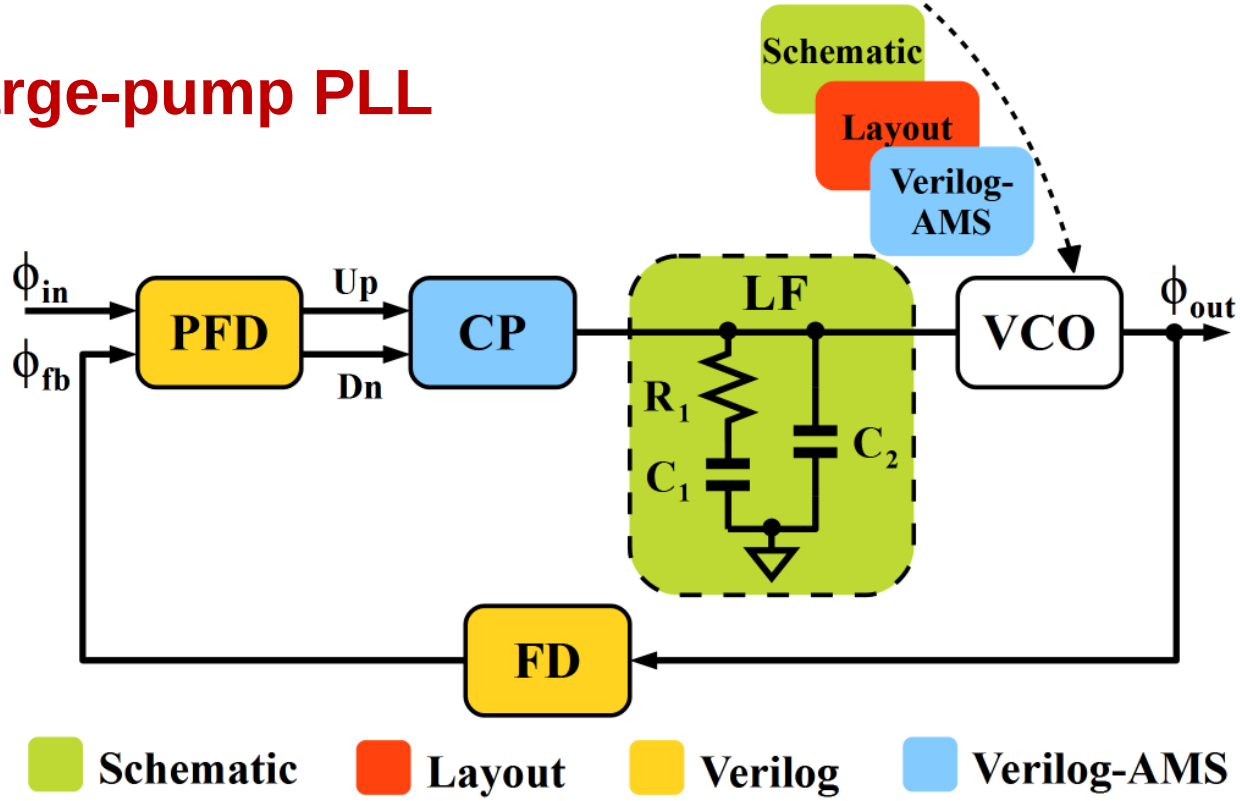
- **Metamodeling can overcome the disadvantages**



- **Model complexity ≠ circuit complexity**
- **Layout-accurate simulation**
- **Suitable for design exploration**

Case Study: PLL Optimization

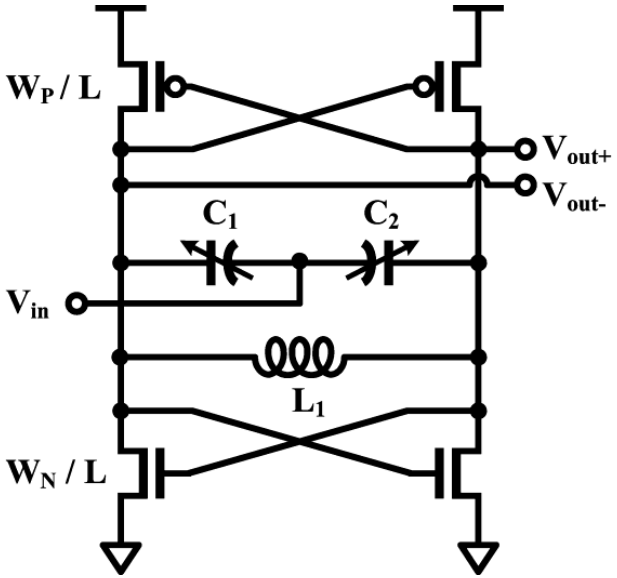
- A VCO Verilog-AMS metamodel is proposed to assist fast PLL design exploration



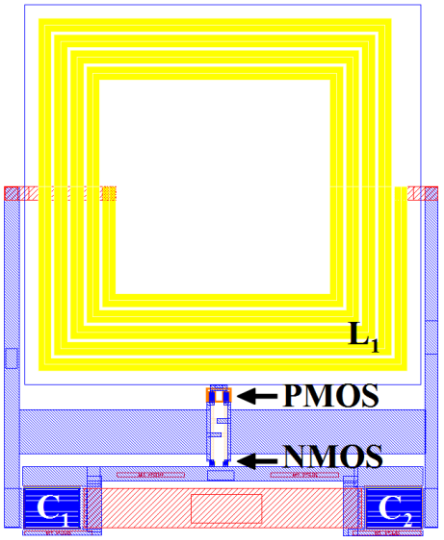
LC VCO Polynomial Metamodel

- A polynomial metamodel is constructed for the 180 nm LC VCO

Schematic



Layout



Polynomial Metamodel

$$f(X)=\sum_{i=0}^{K-1}\beta_i x_1^{P_{1i}} x_2^{P_{2i}} x_3^{P_{3i}}$$

$$x_1 \rightarrow W_P$$

$$x_2 \rightarrow W_N$$

$$x_3 \rightarrow V_{in}$$

VCO Model for Design Exploration

- **The linear model is not suitable for design exploration**

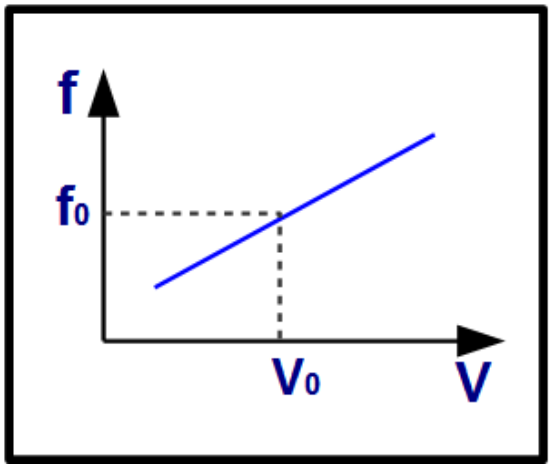
Polynomial Metamodel

$$f(X) = \sum_{i=0}^{K-1} \beta_i x_1^{P_{1i}} x_2^{P_{2i}} x_3^{p_{3i}}$$

Linear model

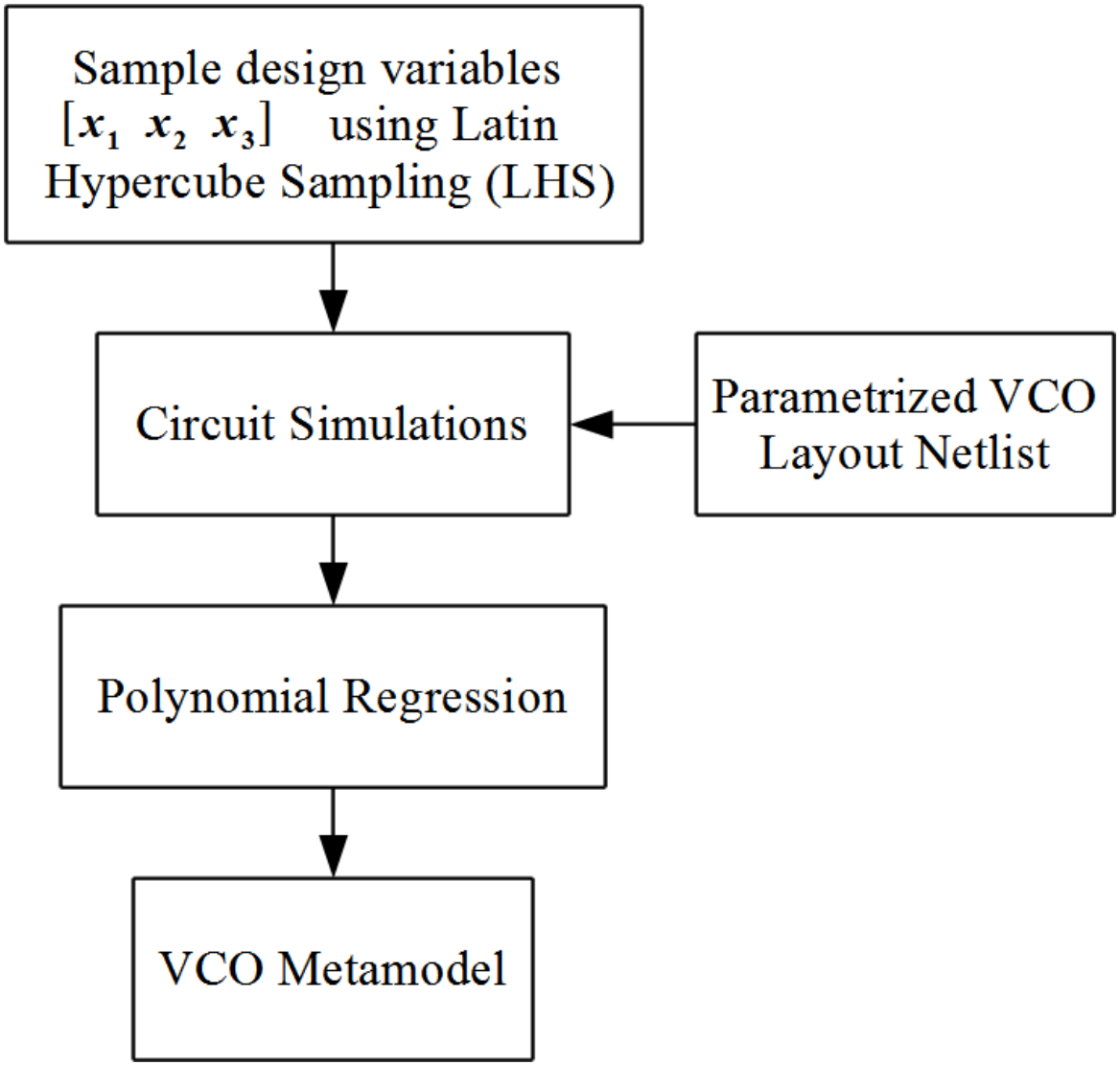
$$f_{osc} = f_0 + K_{VCO} V_C$$

VCO Transfer Curve



Polynomial Metamodel Generation

VCO metamodel generation flow



Verilog-AMS Integration

$$f(X) = \sum_{i=0}^{K-1} \beta_i x_1^{P_{1i}} x_2^{P_{2i}} x_3^{p_{3i}}$$

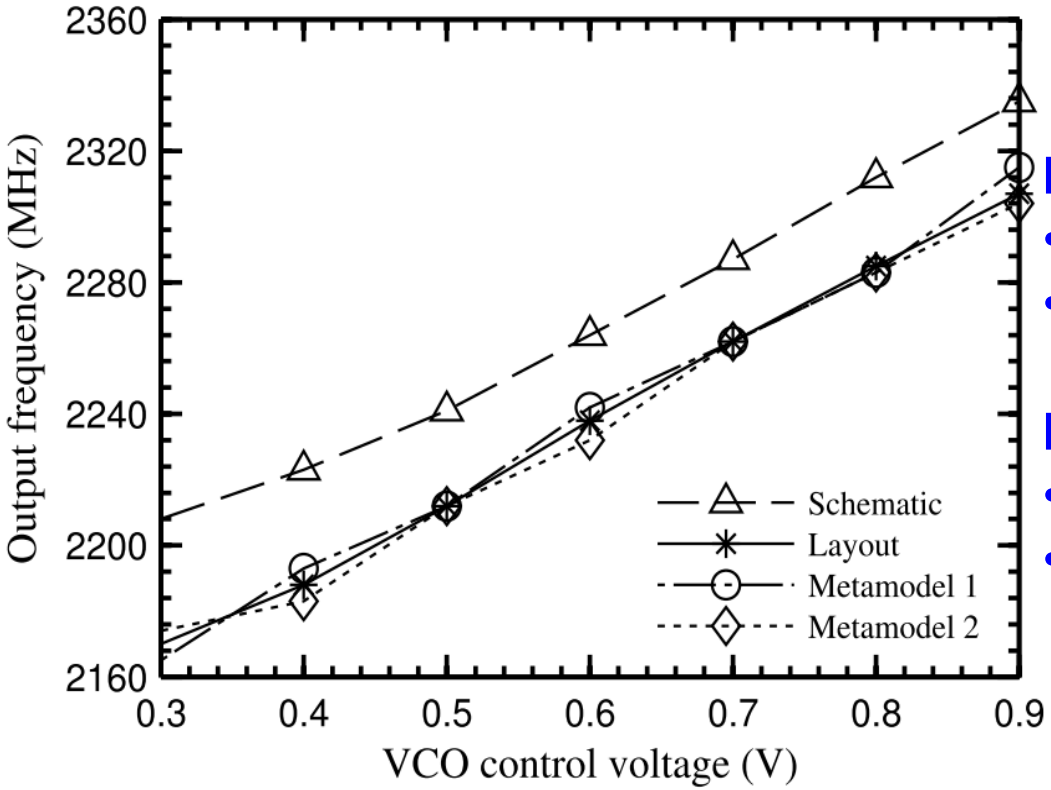
i	p_{1i}	p_{2i}	p_{3i}	$\beta_{i,f}$	$\beta_{i,p}$
0	0,	0,	0,	2.113e+009,	1.385e-005
1	1,	0,	0,	-3.214e+012,	44.459e+000
2	2,	0,	0,	3.456e+016,	-2.804e+005
3	0,	1,	0,	6.869e+012,	39.729e+000
4	1,	1,	0,	-1.021e+017,	2.911e+005
5	0,	2,	0,	-2.071e+017,	-1.080e+006
6	0,	0,	1,	3.513e+008,	-8.271e-004
7	1,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

```
module vco_metamodel (out, in);
... ...
parameter integer K;
initial
begin
... ... // Initialize coefficients
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
```

Verilog-AMS Metamodel

VCO Transfer Curves

- **Quadratic polynomial metamodel is sufficient for modeling the LC VCO**

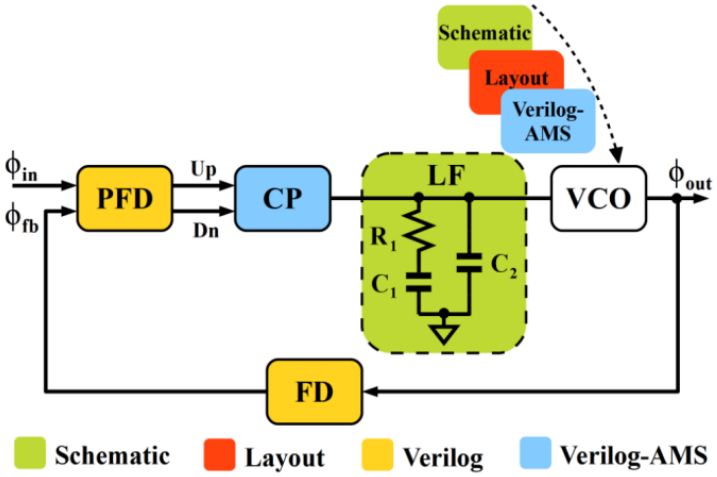


Metamodel 1:
- **100 samples**
- **quadratic model**

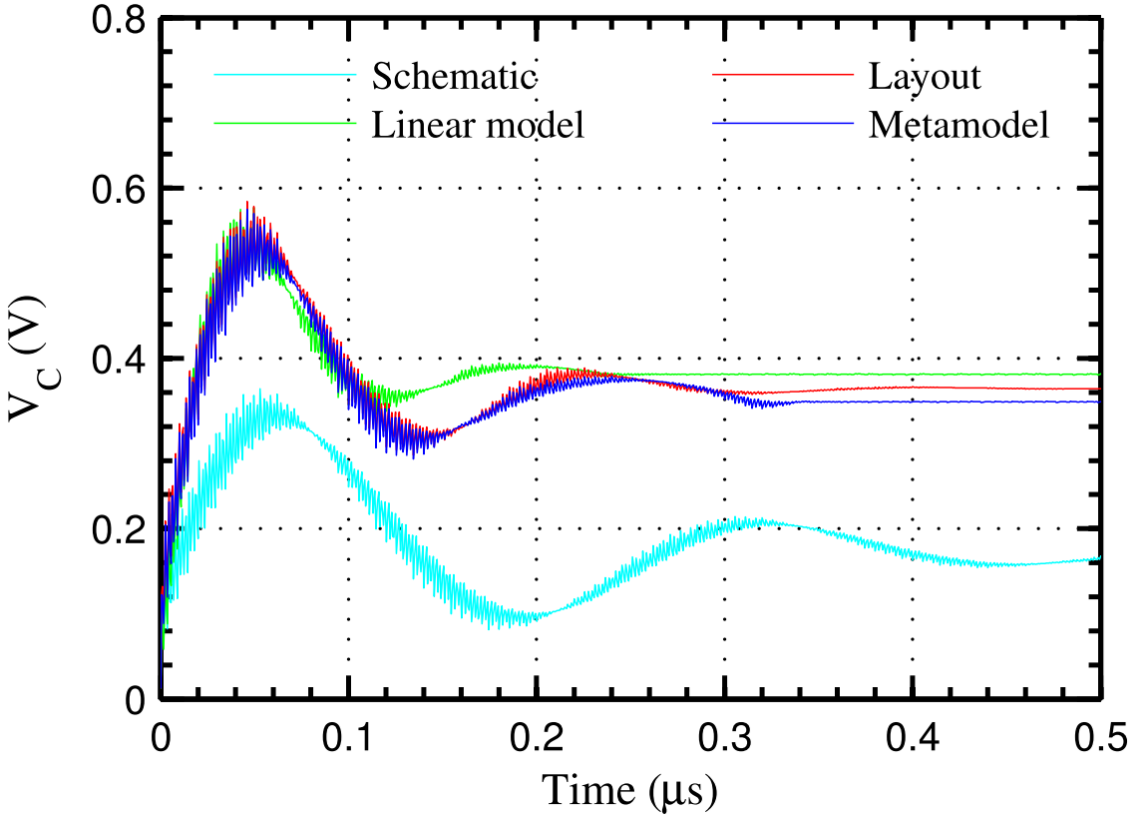
Metamodel 2:
- **500 samples**
- **5-th degree polynomial model**

PLL Behavior with VCO Metamodel

- **The metamodel accurately approximates the PLL behavior**



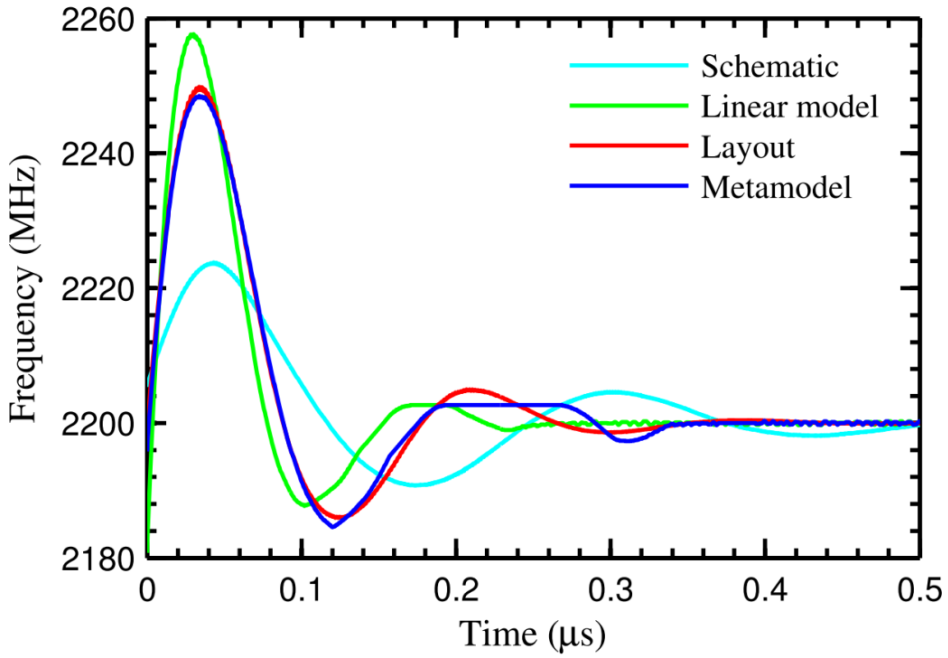
VCO control voltage



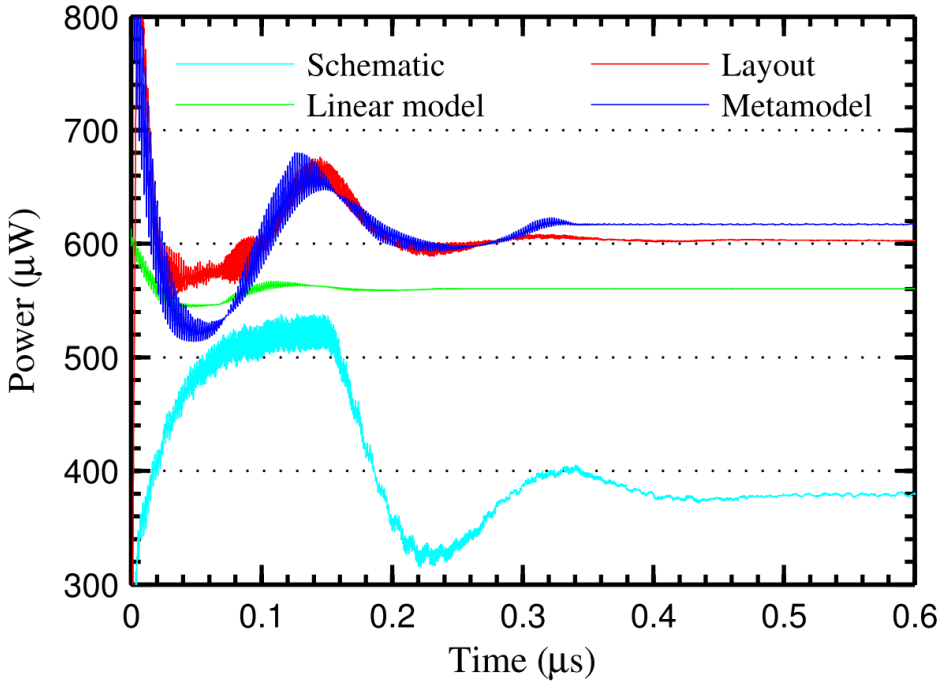
Layout-Accurate Simulation

- **The metamodel is capable of layout-accurate PLL output frequency and power dissipation estimation**

Frequency



Power dissipation



Simulation Speed Comparison

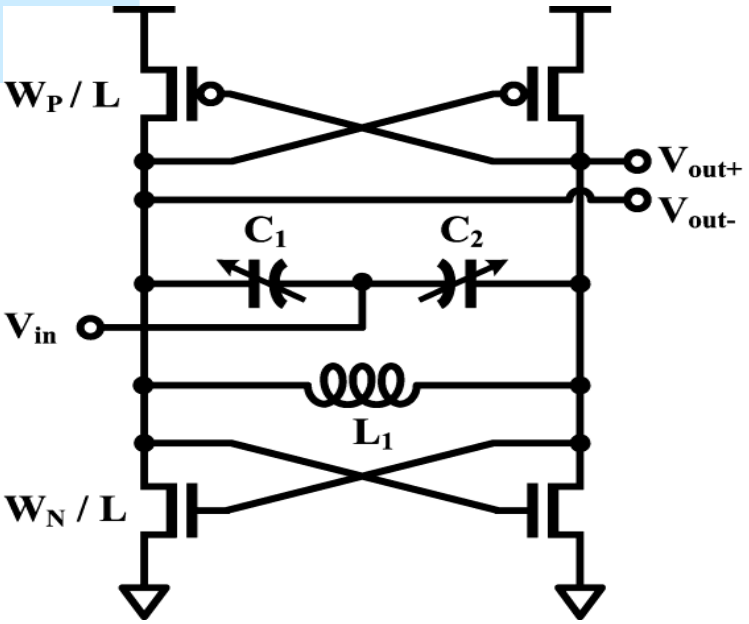
- **10x simulation speedup is achieved using metamodel compared to the one using layout view**

Computation Time Comparison

	Layout	Schematic	Metamodel
Runtime	80.5 s	40.3 s	8.7 s
Normalized speed	1×	∼ 2×	∼ 10×

PLL Optimization

Objective	Minimize lock time, power dissipation	
Constraint	Tuning Range	2180—2300 MHz
Design Variable	W_N W_P	5—15 μm 10—30 μm



PLL Optimization Flow

Metamodel Assisted Exhaustive Search Optimization

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 consid-eration	→	1
5	Verify the final design with layout simulation	→	Done

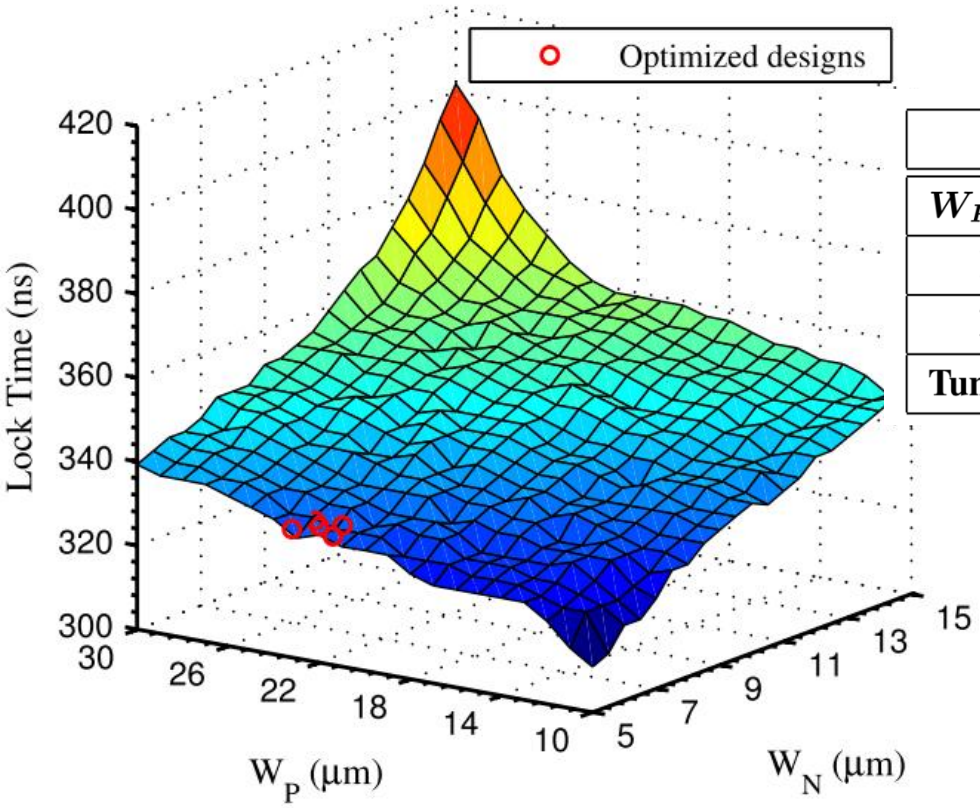
Frequency Metamodel

$$f_{min} \leqslant f \leqslant f_{max}$$

PLL simulation with Verilog-AMS metamodel computation time ~ 30 mins

Optimal VCO design

- **Optimal VCO design is found using the metamodel assisted optimization algorithm**

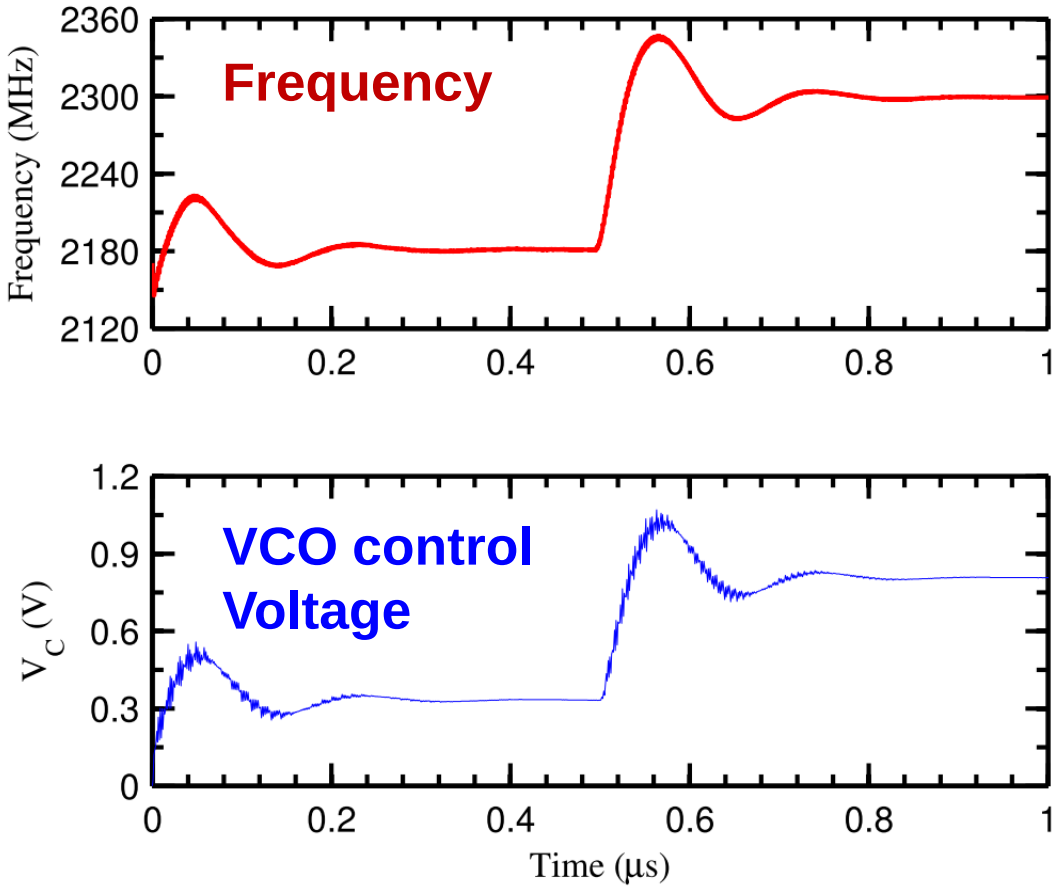


	Baseline	Optimal	Reduction
W_P/W_N (μm/μm)	20 / 10	21.4 / 5	–
Lock time (ns)	335.4	320.4	15.0
P_{Locked} (μW)	602	455	147
Tuning Range (MHz)	2170–2304	2173–2321	–

Sizing the transistors for minimized lock time

Final Design Verification

- **The final design is verified with layout simulation**



PLL first locks to 2180 MHz and then relocks to 2300 MHz

Conclusions

- A Verilog-AMS behavioral model for a 180 nm CMOS LC VCO based on a quadratic polynomial metamodel has been proposed
- Using the VCO metamodel to perform fast and accurate PLL optimization has been demonstrated
- The metamodel can be further improved but is sufficient for lock time and average power estimation
- Parasitic-aware metamodels can be developed for the rest of the PLL building blocks

Questions?

Thank You!!!