Introduction to AMS Behavioral Modeling in SOC

Prof. Chien-Nan Liu
Institute of Electronics
National Chiao-Tung Univ.

Tel: (03)5712121 ext:31211
E-mail: jimmyliu@nctu.edu.tw
http://mseda.ee.nctu.edu.tw/jimmyliu

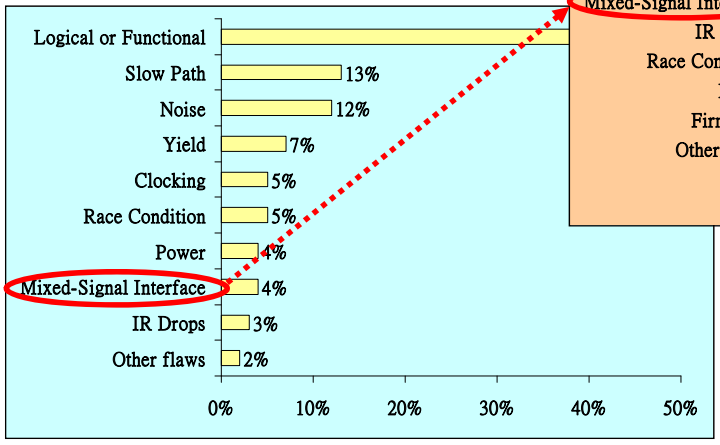
Outline

- AMS circuits in SOC
- Behavioral modeling for analog circuits
- Applications of analog models
  - Noise interactions in AMS systems
    - Supply noise aware behavioral modeling
    - *SCORE* macromodel
  - Yield Enhancement
    - Analysis of process variation effects
    - Process variation aware behavioral modeling

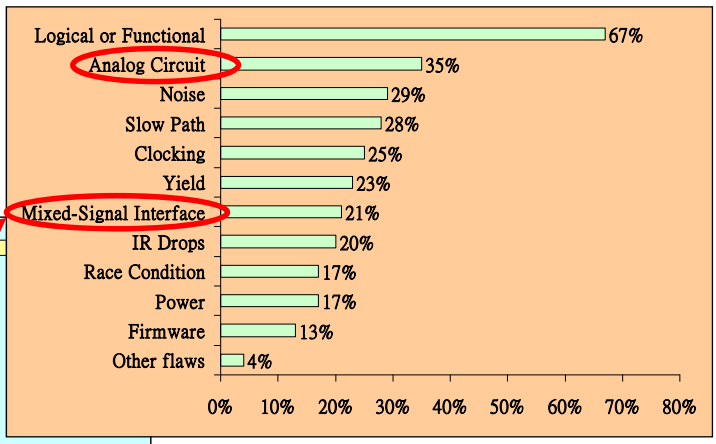
AMS Blocks in SOC

- Analog/Mixed-Signal (AMS) definitions
  - Analog: designs contain **continuous signal**
    (ex. continuous time filter, OP amp., mixer…)
  - Mixed-signal: designs contain **both** analog and digital signals (ex. A/D, D/A, PLL…)
    - Mainly focused on analog functionality
- SOC designs often include the analog interface to the outside world
- Synthesis still does not exist for AMS blocks
  - AMS languages only model the **behavior** of AMS blocks
  - Today, AMS VCs are in **hard (layout) form**

Major Respin Causes

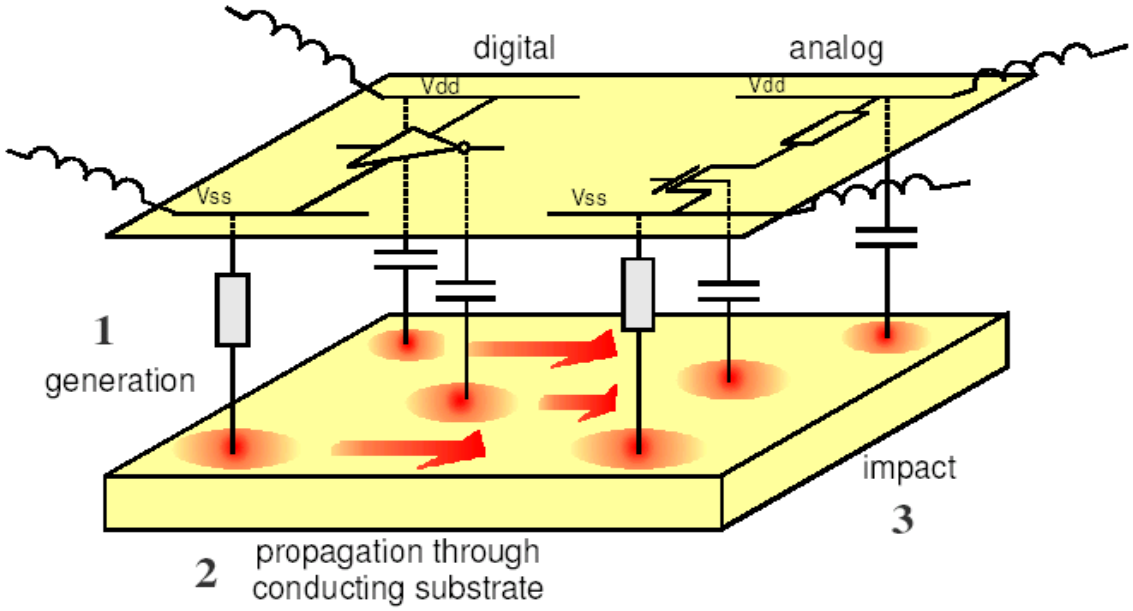


Collett Intl. 2000 Survey



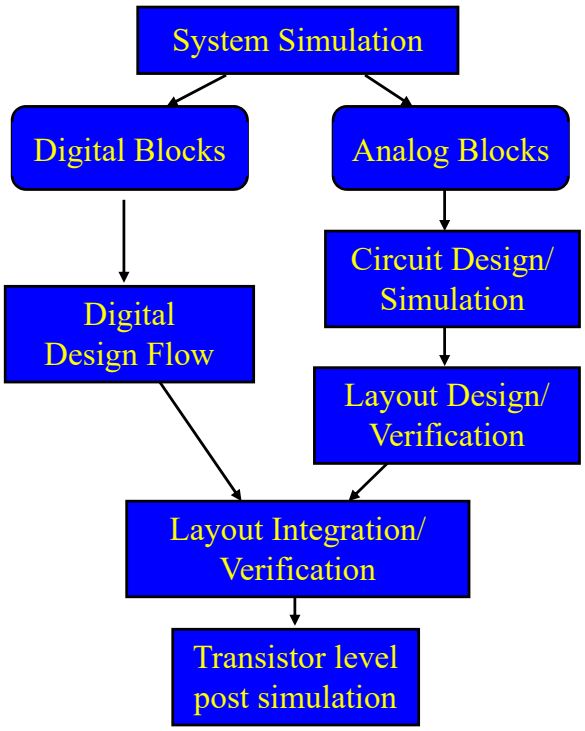
Collett Intl. 2001 Survey

Noise Coupling in AMS Designs



Source: Workshop on Substrate Noise-Coupling in Mixed-Signal ICs, Imec, Belgium, Sep. 2001

Conventional MS Design Approach



- **Design/simulate digital and analog circuits separately**
  - May miss the interaction effects
- **Can perform co-simulation at transistor level only**
  - High complexity
  - Too slow
- **Solutions**
  - Use a systematic, top-down design approach to capture design intent
  - Develop some tools to rapidly target for different requirements

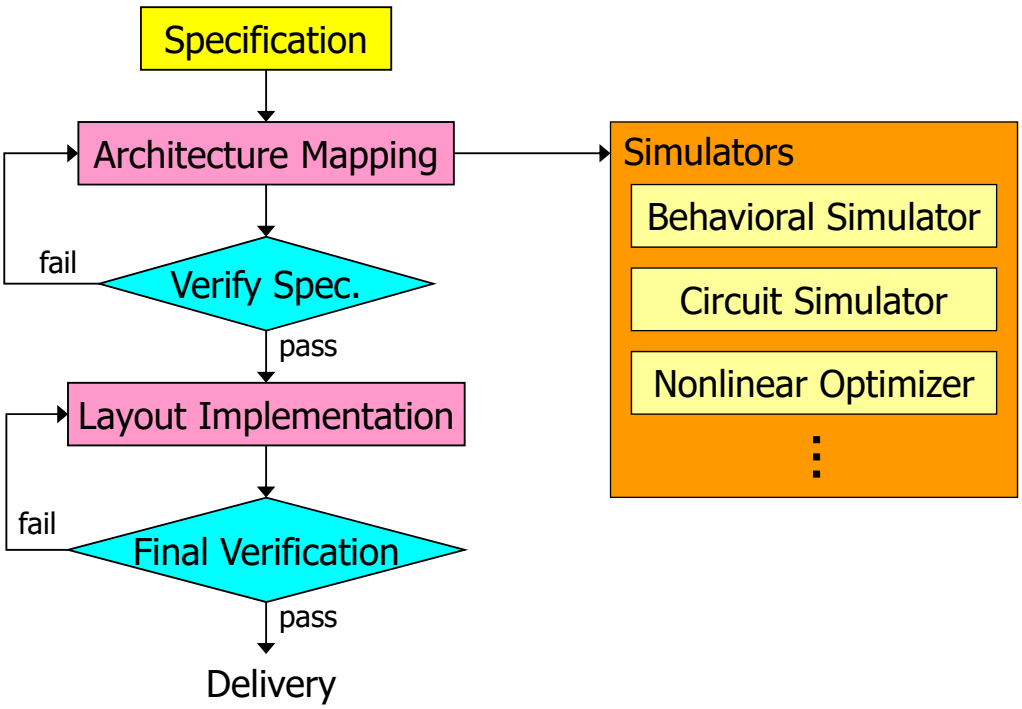
Traditional Analog Simulation

- Analog designers have been using **SPICE** or SPICE-like tools for analog simulation over 30 years
  - Contain models of circuit elements (R, L, C, …)
- Perform various analysis of circuits with high accuracy
  - DC, AC, transient, TF, …
- Limited to small circuits due to long computation time
- Existing fast spice products are not still suitable for nanometer circuit analysis
  - Insufficient accuracy caused by simplified model
  - Latency assumption fails to address logically idle but electrically active nature of nanometer circuit behavior

Top-Down AMS Design Flow

- Starting from **behavioral models** to check system behavior

決定電路架構與sizing
--> pre-layout



Adv. Of Top-Down Methodology

- A widely accepted concept for digital designs
- Start design by using **behavioral modeling**
- Allow system simulation and architecture verification
- Allow design and verification of the circuit architecture before block design
- Allow mixed-level simulation with other digital circuits
- Allow changes with minimum impact to the design cycle

Outline

- AMS circuits in SOC
- Behavioral modeling for analog circuits
- Applications of analog models
  - Noise interactions in AMS systems
    - Supply noise aware behavioral modeling
    - *SCORE* macromodel
  - Yield Enhancement
    - Analysis of process variation effects
    - Process variation aware behavioral modeling

Analog Behavioral Modeling

- A mathematical model written in **Hardware Description Language (HDL)**
  - Verilog-AMS
  - VHDL-AMS
  - Matlab
  - C/C++
  - ……
- Emulate circuit block functionality by sensing and responding to circuit conditions
  - Simulate at **behavioral level**
- **Faster** simulation time
  - Allow **whole chip simulation**

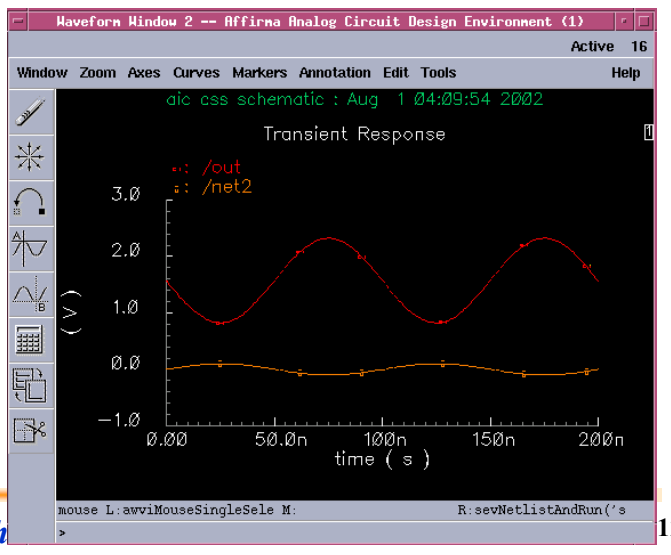
An Example of Verilog-AMS Code

- **Keys to a good behavioral model**
  - Concise mathematical equations of the behavior
    - For faster simulation time
  - Appropriate value for each parameter
    - For accurate simulation results

Text Editor V3.6.2 FCS [cae30] - veriloga.va

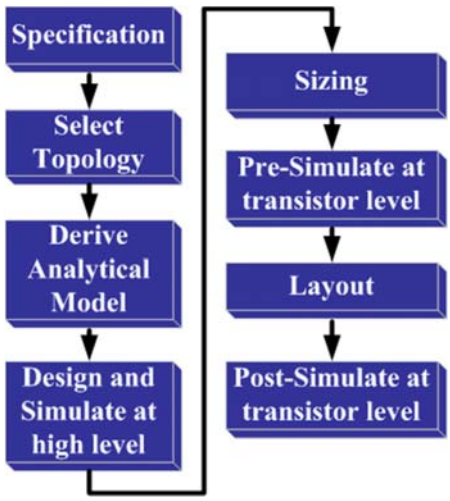
File View Edit Find

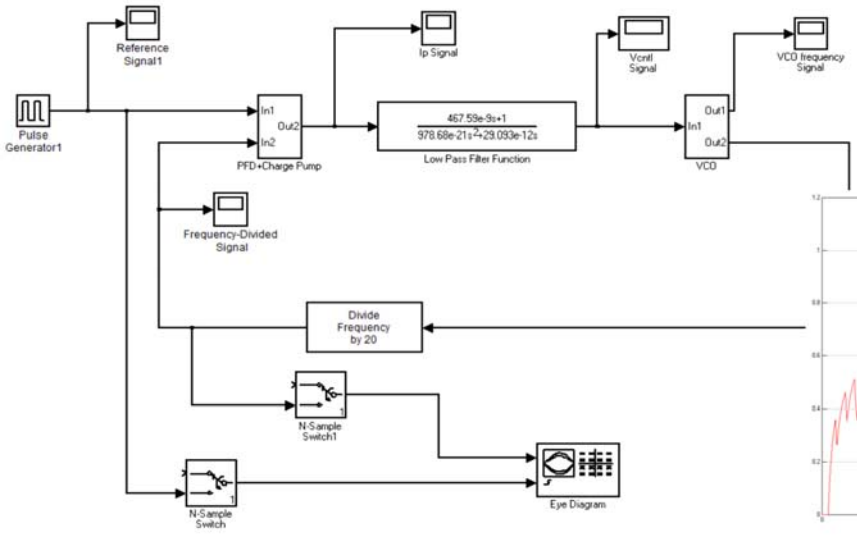
```
// VerilogA for aic, cs, veriloga

`include "constants.h"
`include "discipline.h"

module cs(in, out);
input in;
output out;
electrical in, out;

parameter real gain=-7.5;
parameter real vdd=3.3;
parameter real id=0.23m;
parameter real rd=7.5e3;

 analog begin

  V(out) <+ (gain * V(in))+vdd-(id*rd) ;
 end
endmodule
```

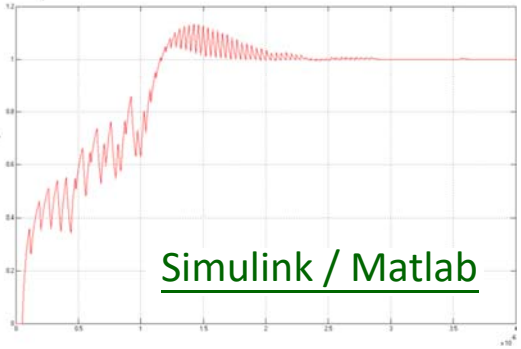


Top-down Modeling Approach

- **Top-down approach** can verify whole AMS system **roughly** before implemented
  - Suitable for **newly-created designs**
- **Issues:**
  - Lack of layout information (loading, parasitic, …)
  - Inaccurate on non-ideal circuit properties





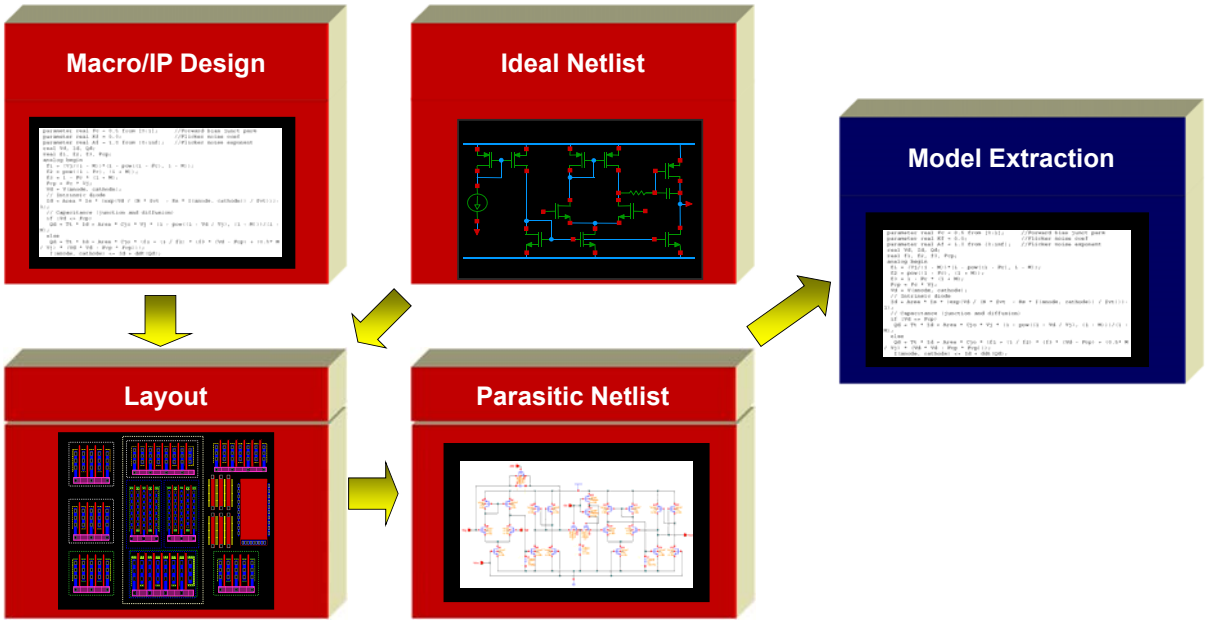


Simulink / Matlab

Bottom-Up Modeling Approach

- While using ***existing*** blocks, bottom-up ***behavior extraction*** is required for system verification
  - An interesting research direction
- Bottom-up approach can be much accurate
  - More accurate (loading effects, parasitic effects, …)
  - Actual non-ideal circuit information
  - Signal interaction effects
- Bottom-up approach can still effective when those design parameters are hard to obtain
  - Only have flattened transistor-level design
  - Suitable for IP-based designs (SOC designs)

Bottom-Up Behavior Extraction



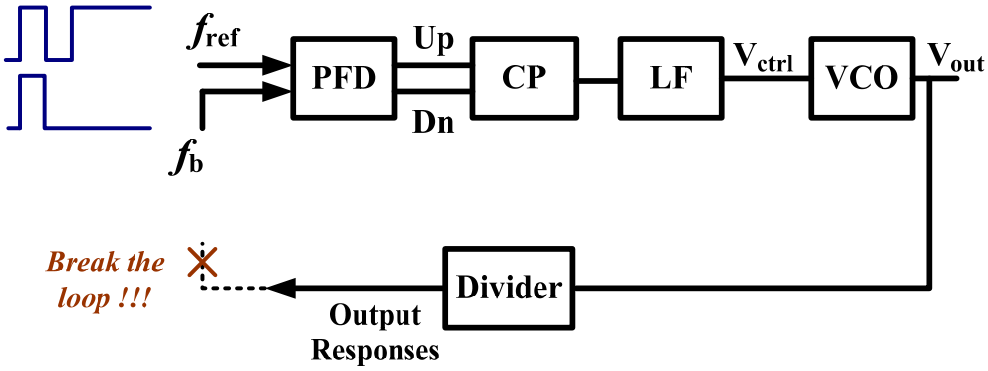
Accurate Behavior Extraction

- **Bottom-up extraction from simulation results**
  - More accurate
  - Still useful for flattened designs
- **Do not separate into sub-blocks**
  - Correctly deal with timing information, loading, parasitics and interactions
- **Do not measure from normal operations**
  - Develop a **special characterization mode**
  - Easily send special patterns to trigger the circuit
  - Long extracting time can be avoided

Characterization Mode

Example: Charge Pump Phase-Locked Loop (CPPLL)

(Extraction patterns)

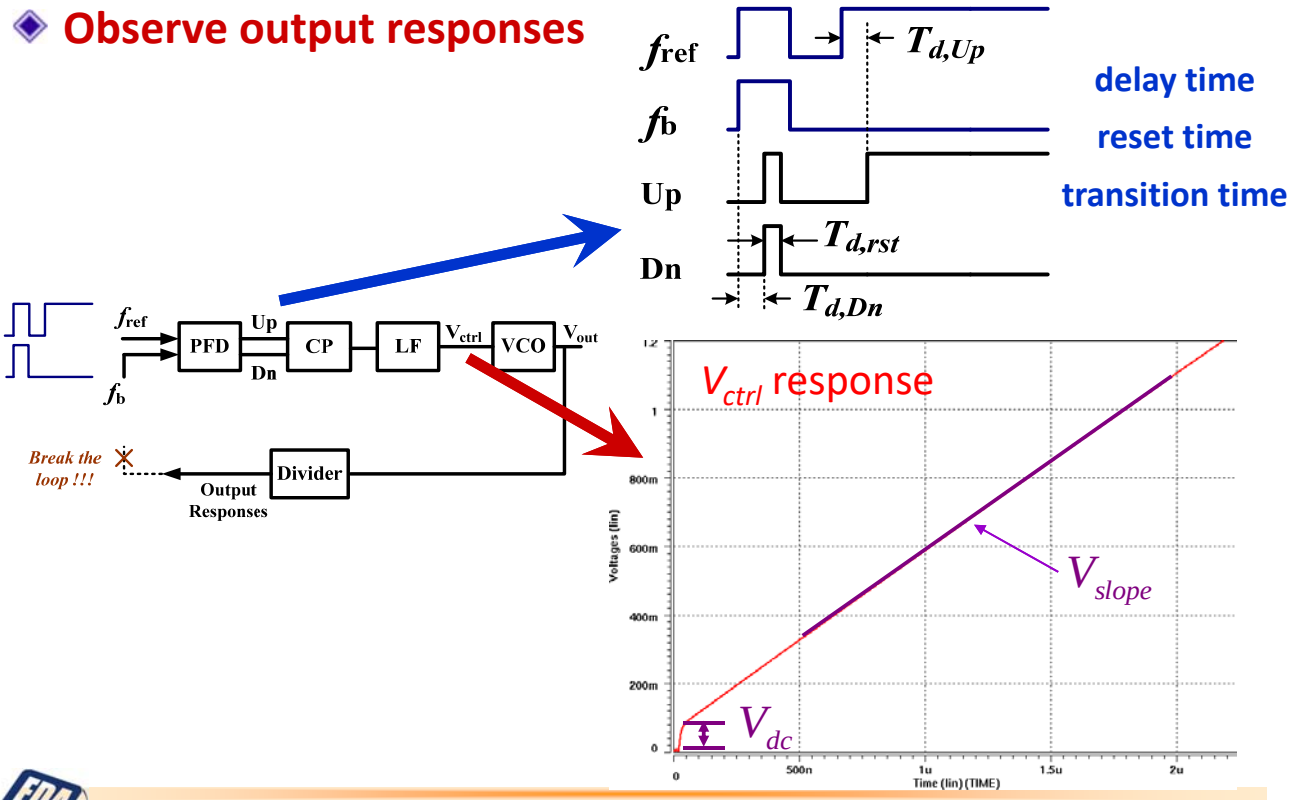


C.C. Kuo, Y.C. Wang, and **C.N. Liu**, "An Efficient Approach to Build Accurate PLL Behavioral Models of PLL Designs", *IEICE Trans. on Fundamentals (SCI)*, vol. E89-A, no. 2, pp. 391-398, Feb. 2006.

C.C. Kuo, Y.C. Wang, and **C.N. Liu**, "An Efficient Bottom-Up Extraction Approach to Build Accurate PLL Behavioral Models for SOC Designs", *ACM/IEEE GLSVLSI*, pp. 286-290, Apr. 2005.

Extract Circuit Properties

Observe output responses



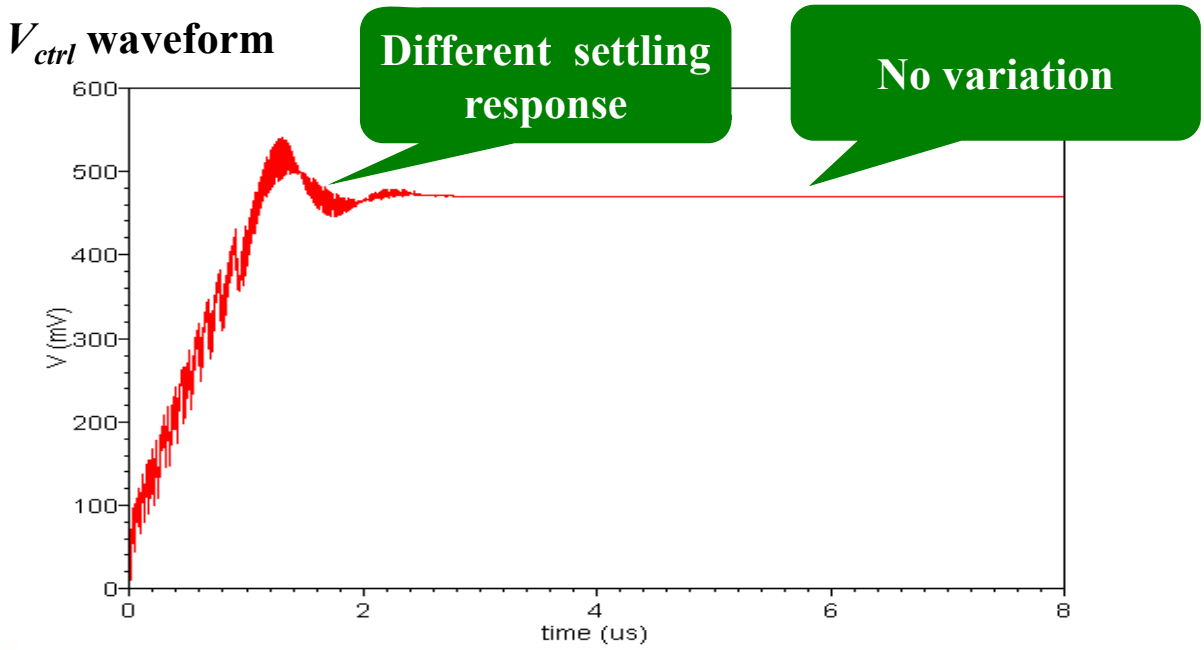
Case Study

- Charge pump PLL
- Use Verilog-AMS language to describe PLL behaviors
- **Simulation environment**:
  - Analog Artist (Cadence)
  - Simulator: Spectre

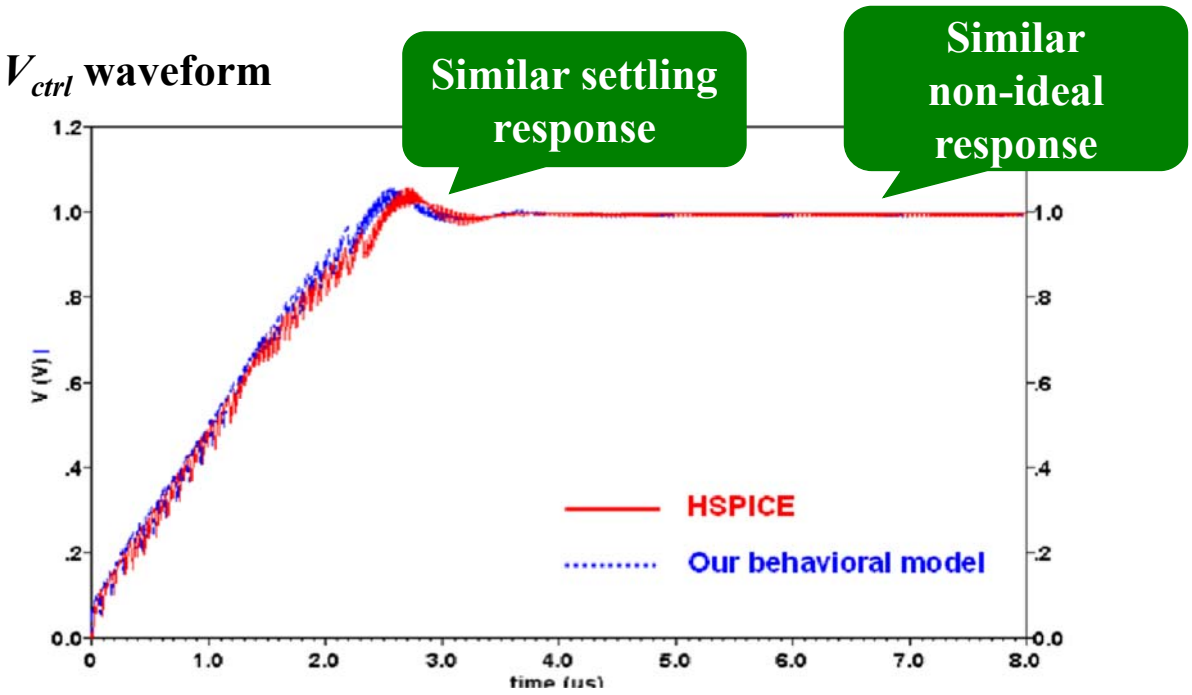
	Specification
Process	TSMC 0.18um
Input freq.	25MHz
Output freq.	800MHz
T_{lock}	< 5us
pk-pk Jitter	< 20ps

Ideal Behavioral Model

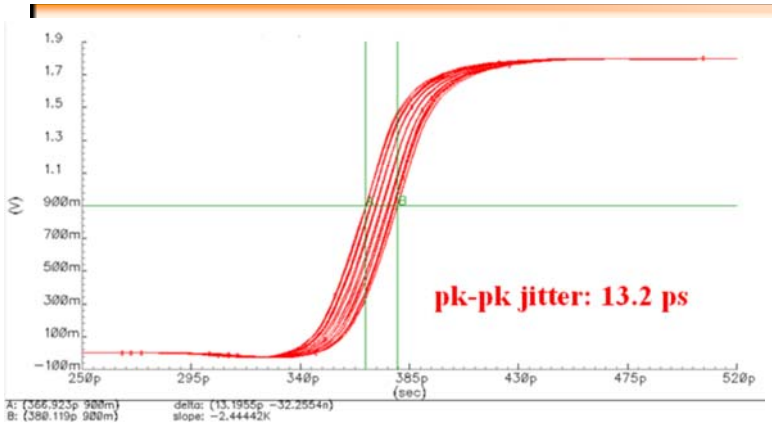
- Use the embedded behavioral blocks from **Cadence's AHDL library**



Extracted Behavioral Model

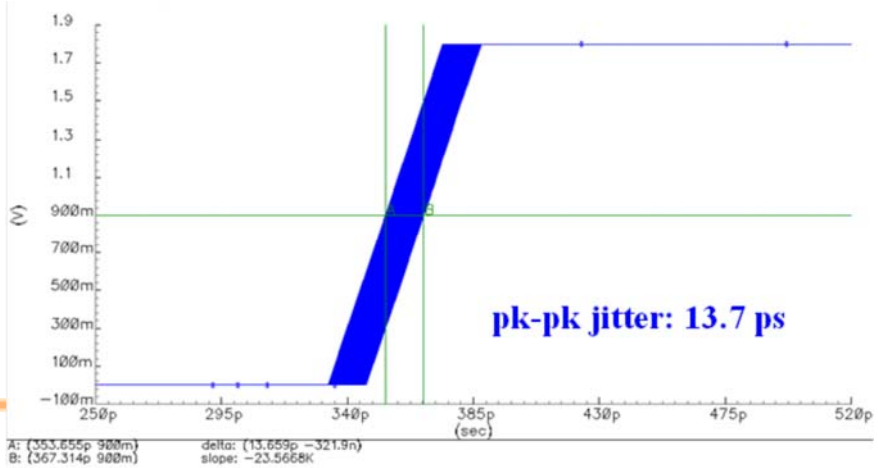


Peak-to-peak Jitter

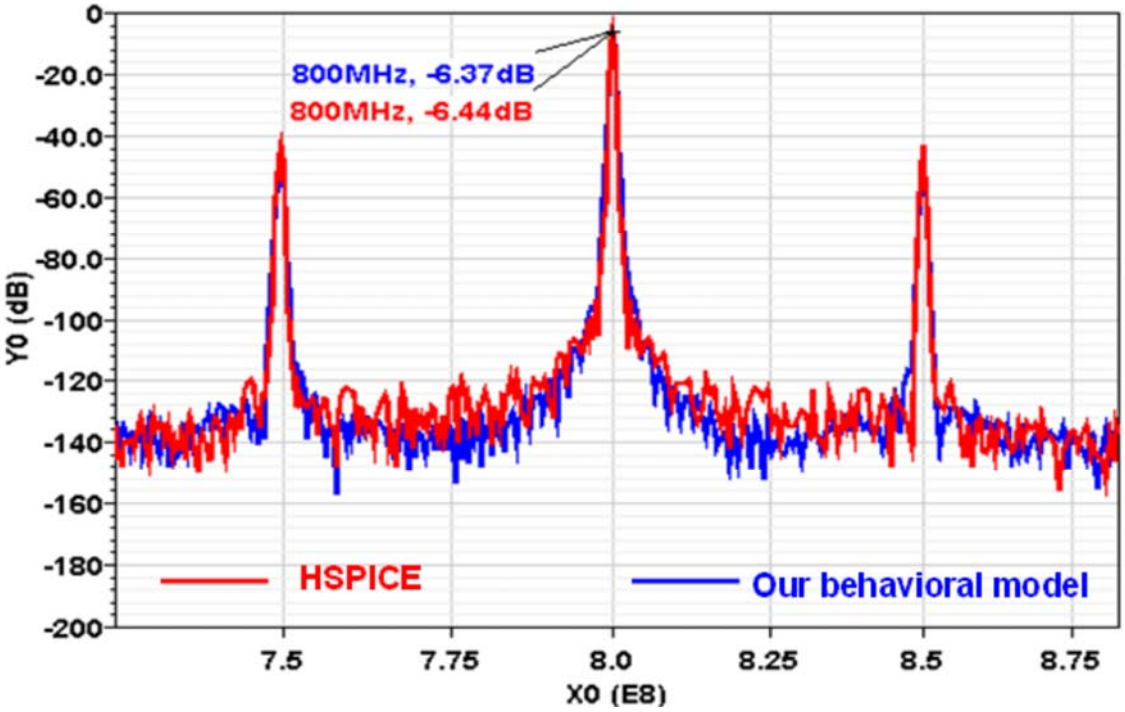


HSPICE : 13.2 ps
$T_{sim.}$: 22074 sec

Beh model: 13.7ps
$T_{sim.}$: 122 sec (180x)



Similar Frequency Responses



Outline

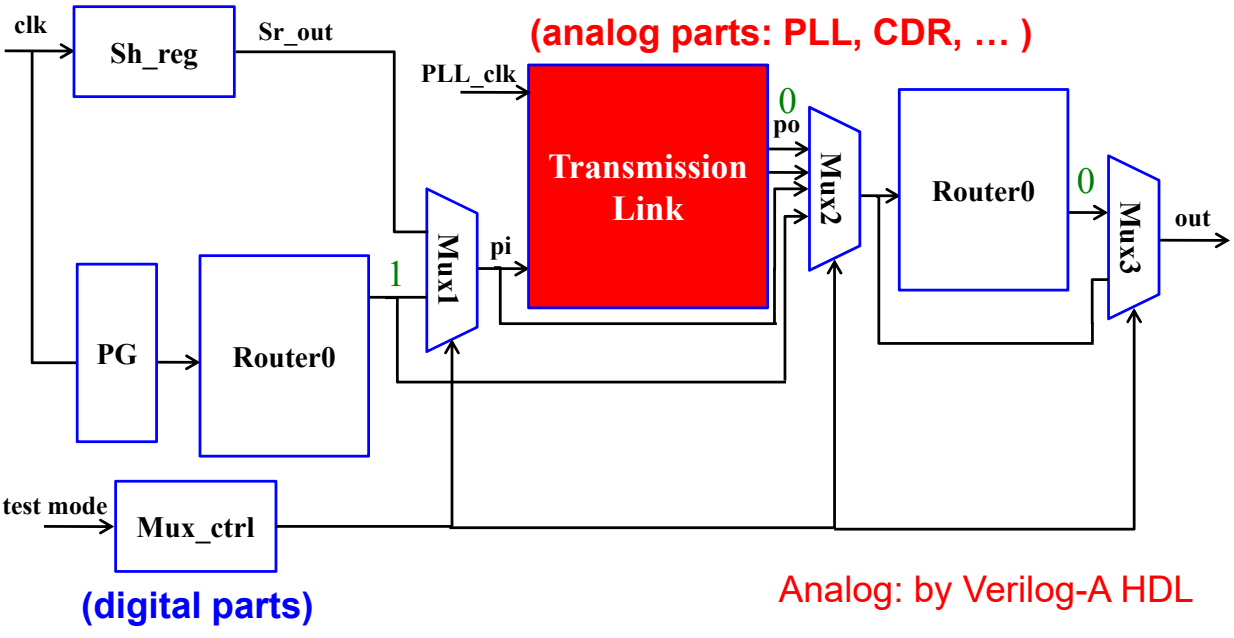
- AMS circuits in SOC
- Behavioral modeling for analog circuits
- Applications of analog models
  - Noise interactions in AMS systems
    - Supply noise aware behavioral modeling
    - *SCORE* macromodel
  - Yield Enhancement
    - Analysis of process variation effects
    - Process variation aware behavioral modeling

Possible Applications

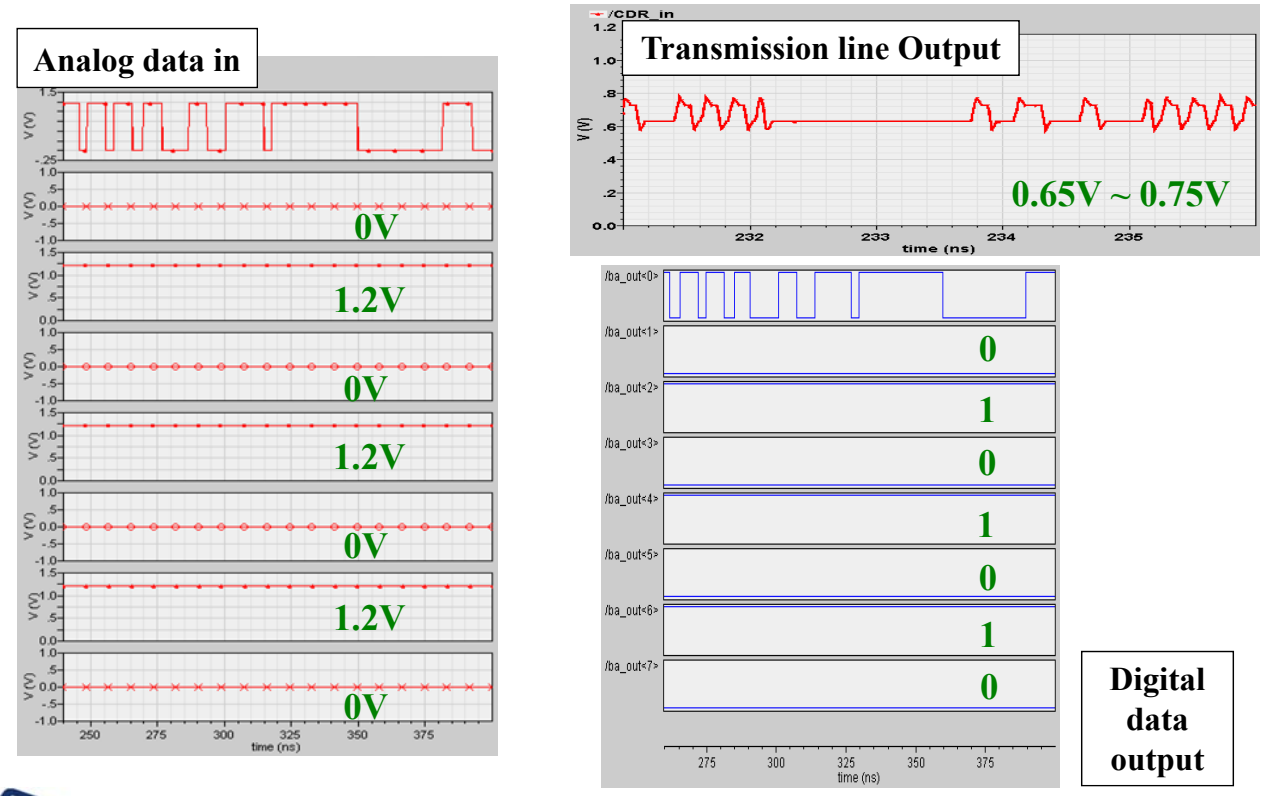
- Using analog behavioral models has fast simulation time
  - Can be used to replace the time-consuming simulation process in traditional design flow
  - Model accuracy is the key issue to be solved
- Used for system simulation with digital/analog circuits
  - Digital circuits are too large to be simulated by HSPICE
- Used for analyzing the noise effects in analog circuits
  - Noise-aware behavioral models can help to check noise issues
- Used for analyzing the design yield of analog circuits under process variation
  - Fast simulation time can speedup the Monte Carlo simulation
  - Enable designers to make improvement at behavioral level

Analog Models in System Design

- **Transmission link system**



AMS System Simulation



Simulation Time : 2727 sec (5000 data)

Outline

- AMS circuits in SOC
- Behavioral modeling for analog circuits
- Applications of analog models
  - Noise interactions in AMS systems
    - Supply noise aware behavioral modeling
    - *SCORE* macromodel
  - Yield Enhancement
    - Analysis of process variation effects
    - Process variation aware behavioral modeling

Analog Performance Variations

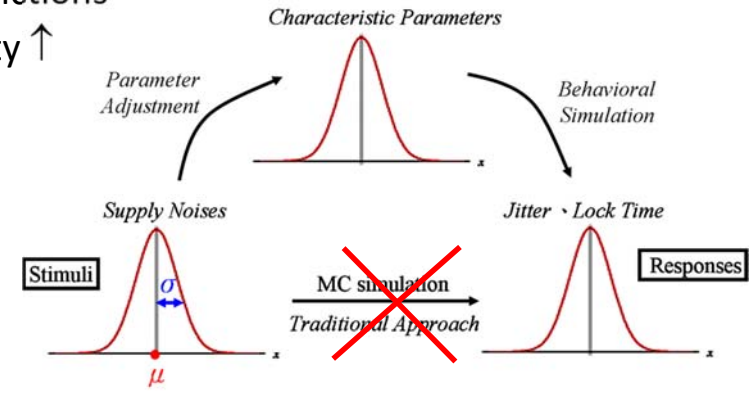
- **PVT** (Process, Voltage, and Temperature) variations have large performance impact
  - **Process variation**
  - **Supply noise**
  - Substrate noise
  - Temperature issue

  } **especially from digital circuits**

- Performance variation analysis of **analog circuits** often needs **expensive transistor-level simulation**
- **Efficient analyzer** is necessary for system design
  - **Variation-aware behavioral modeling approach**

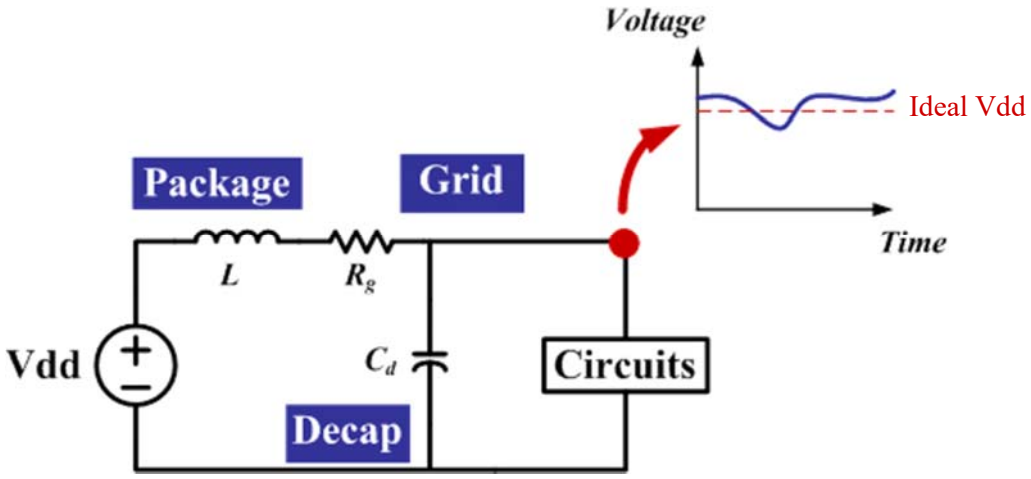
Noise-Aware Behavioral Models

- Process variation and outside noise have large impacts on analog circuits
  - Often appear as random variables
- Typical approach: model the performance (timing, jitter, …) as a function of those random variables
  - Use curve fitting to obtain the parameters of those functions
  - # variables ↑, complexity ↑
- Alternative approach:
  - Use behavioral models
  - Find suitable internal parameters instead of trying to fit the final performance directly



Supply Noise

- Issues
  - $L\,di/dt$
  - IR drop
- Impacts
  - Performance loss
  - Yield loss



Stochastic Analysis Approaches

Random noise

(Gaussian) (Uniform)

Supply noise

↓

Random generator

↓

Circuit performance

Issues:

- Rough prediction
- May be quite different in real cases

Regular Noise Analysis

Known waveform

Supply noise

Regression

or

circuit analysis

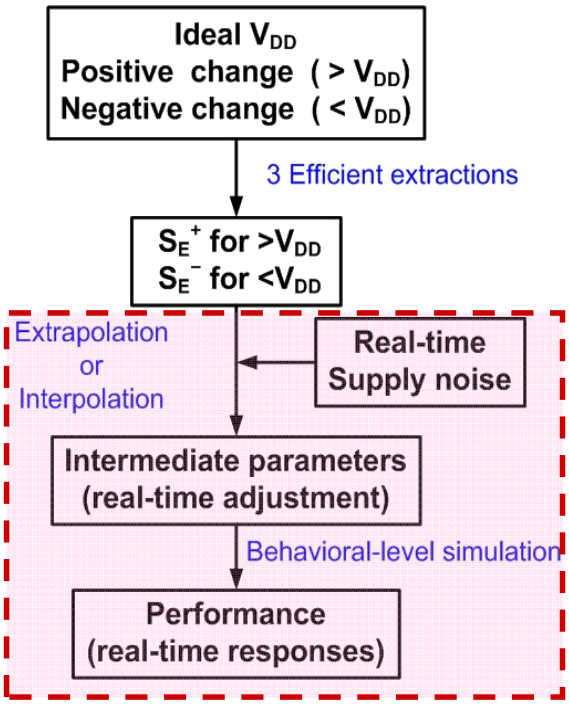
Circuit performance

Issues:

- Need complicated equations
- Not actual supply noise:
  - Unpredictable and Irregular

Handle Irregular Supply Noise

- **Behavioral-level simulation**
  - **Save much simulation time**
- **Dynamic adjustment**
  - **Real-time calculation using current noise status**
- **Suitable for real systems**
  - **Handle unpredictable noise**



C.C. Kuo and **C.N. Liu**, "Accurate Behavioral Modeling Approach for PLL Designs with Supply Noise Effects", IEEE BMAS, pp. 48-53, Sept. 2005.

C.C. Kuo and **C.N. Liu**, "On Efficient Behavioral Modeling to Accurately Predict Supply Noise Effects of PLL Designs in Real Systems", IFIP *VLSISOC*, pp. 116-121, Oct. 2006.

Linear Model for Intermediate Parameters

For example:

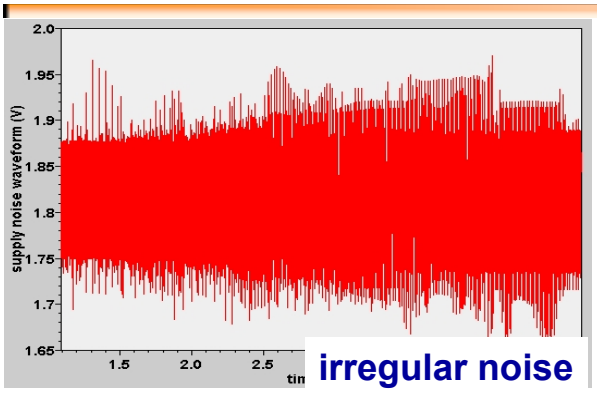
$$T_d + \Delta t = f(V_{DD} + \Delta vdd)$$

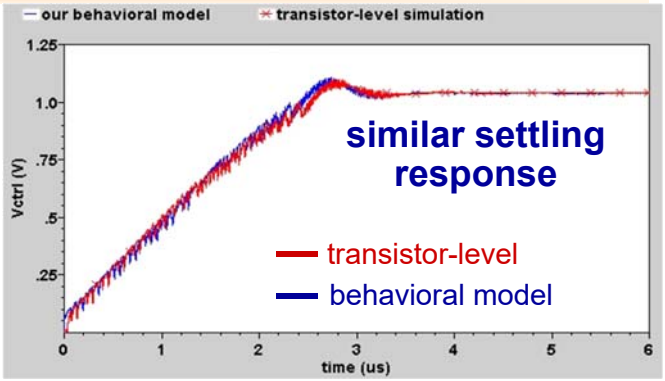
Sensitivity analysis under different V_{DD}:

$$S_E = \frac{\Delta delay}{\Delta V_{DD}} \ (constant)$$

$$\rightarrow \quad S_E = \frac{(T_d + \Delta t) - T_d}{(V_{DD} + \Delta vdd) - V_{DD}} \qquad \begin{pmatrix} S_E^{+} & \text{for } \Delta vdd > 0 \\ S_E^{-} & \text{for } \Delta vdd < 0 \end{pmatrix}$$

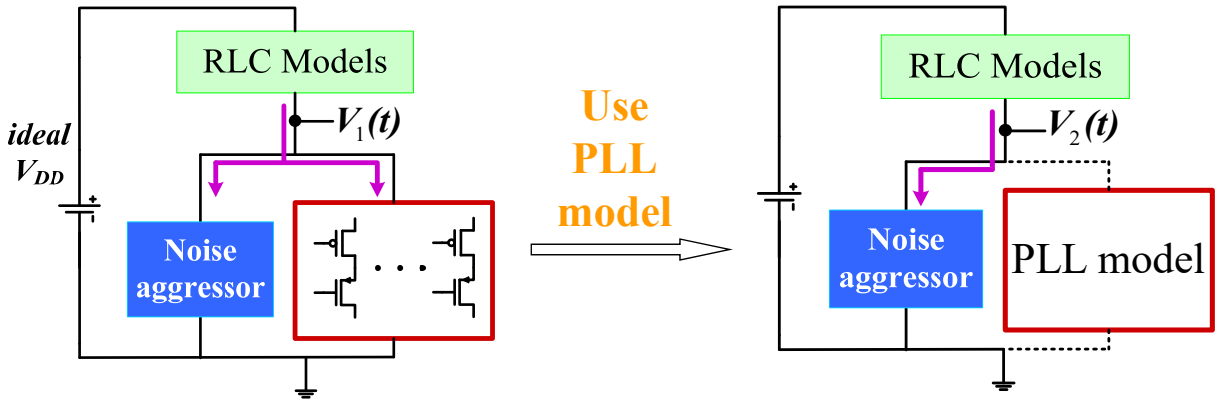
Simulation with Supply Noise





		Circuit-level	Ours	Noise free model
From V_{ctrl} waveform	V_{max} (V)	1.095	1.102	1.083
	V_{lock} (V)	1.0352	1.0348	1.034
	ΔV_{ctrl} (mV)	6.6	6.8	6.1
	pk-pk PJ @ 800MHz (UI)	0.0176	0.016	0.0082
	$T_{simulation}$ (sec)	40413	1507	824

Noise Issues in AMS Integration

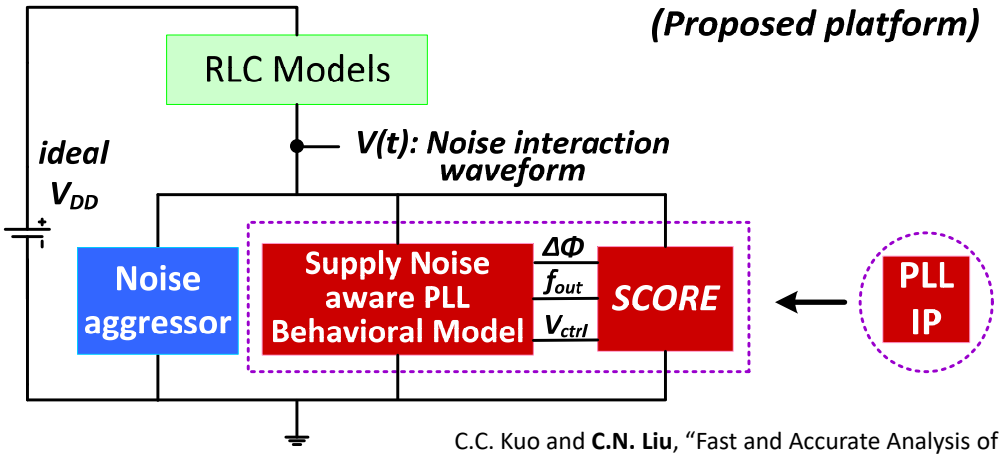


- **Using a noise-aware behavioral model** may not accurate to analyze supply noise effects
  - **V_{DD} waveform is independent on PLL behaviors**
  - **Only** consider the **external supply noise**
  - When the noise affects PLL behaviors, the behavior changes also influence the noise waveform

SCORE Macromodel for Noise Interaction

- ***SCORE***: State-controlled resistors
- **Help PLL models handle supply noise interaction issues**
  - Provide accurate V_{DD} waveforms with PLL interactions
  - Can be combined with other noise-aware approaches

(Proposed platform)

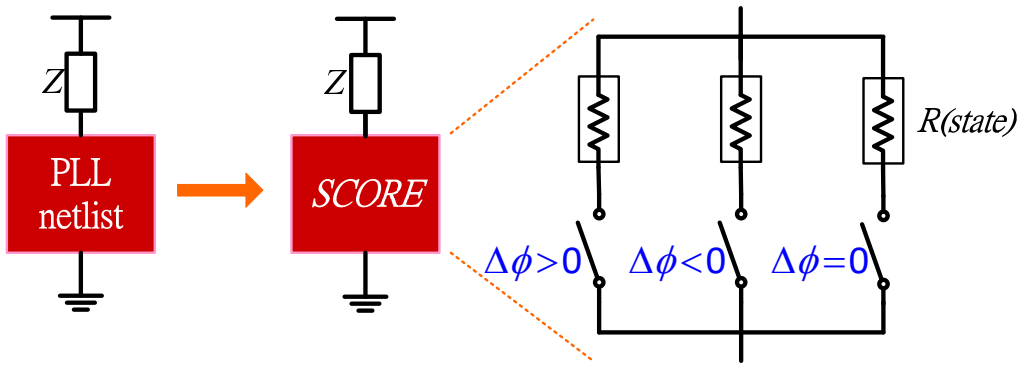


ΔΦ: current phase difference

C.C. Kuo and **C.N. Liu**, "Fast and Accurate Analysis of Supply Noise Effects in PLL with Noise Interactions", *IEEE Trans. on Circuits and Systems I*, vol.57, no.1, pp.44-52, Jan. 2010.

Ideas of *SCORE* Macromodel

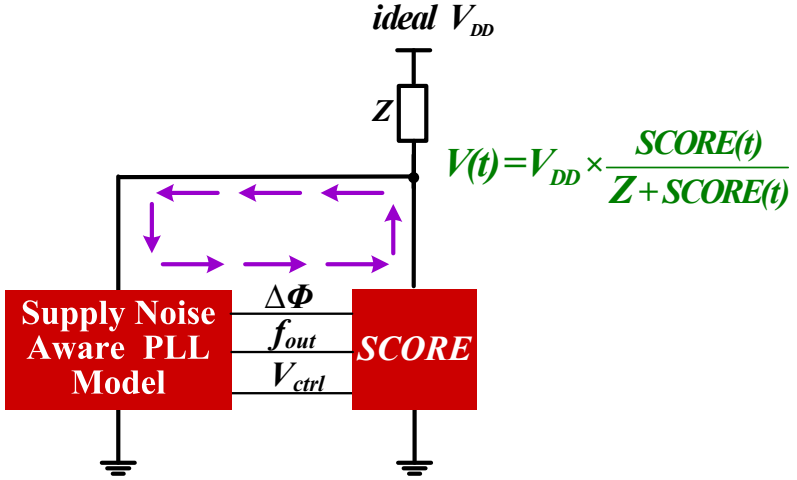
- **Treat whole PLL design as a black-box**
- **Approximate the PLL behaviors** under a parasitic power line
- **Not record entire PLL I_{DD} waveform** to reduce cost
  - Extract the **peak voltages** induced by the PLL $I_{DD,peak}$
- Model the PLL as **state-controlled switches** and **resistors**
  - PLL behaviors **in system view: 3-state machine**



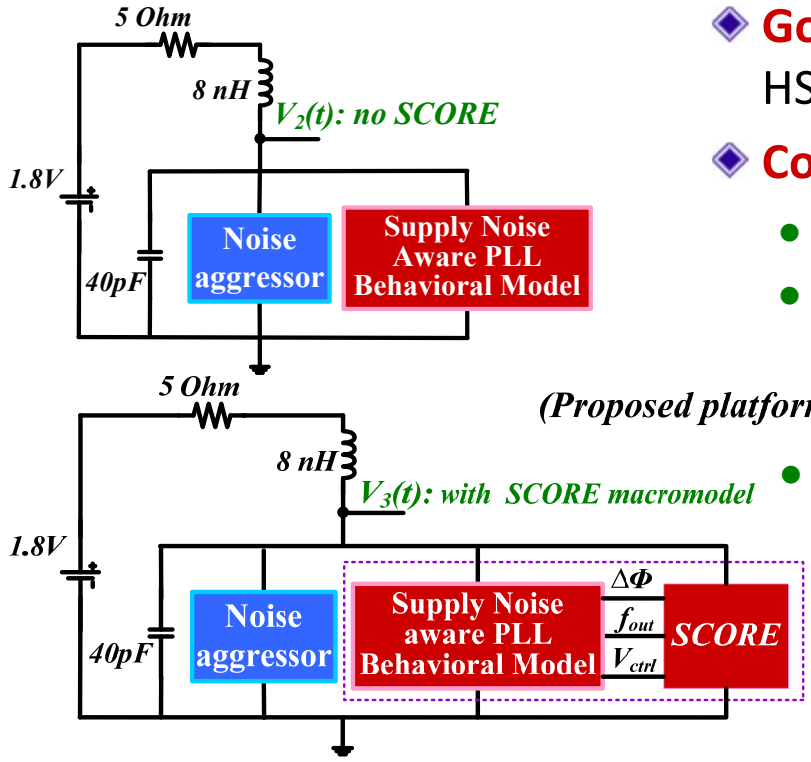
Recursive Simulation Platform for PLL

- The real-time supply noise *V(t)* affects the PLL behaviors
- The PLL behaviors also affect the supply noise *V(t)*
  - **Supply noise is not only an independent input**
  - *V(t)* also varies with real-time circuit behavior

  (just like using a transistor-level simulator)



Experiments

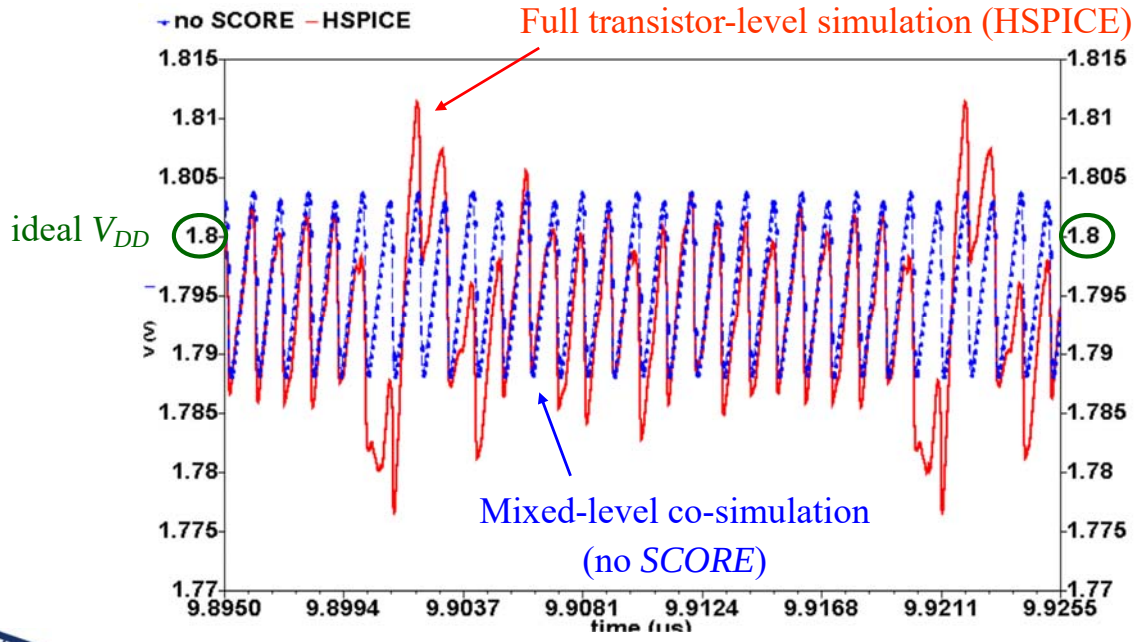


(Proposed platform)

- **Golden reference**: HSPICE simulation
- **Comparison**
  - Supply noise waveform
  - PLL responses under such noise
  - Noise aggressor: 3-stage buffer chain (transistor-level netlist)

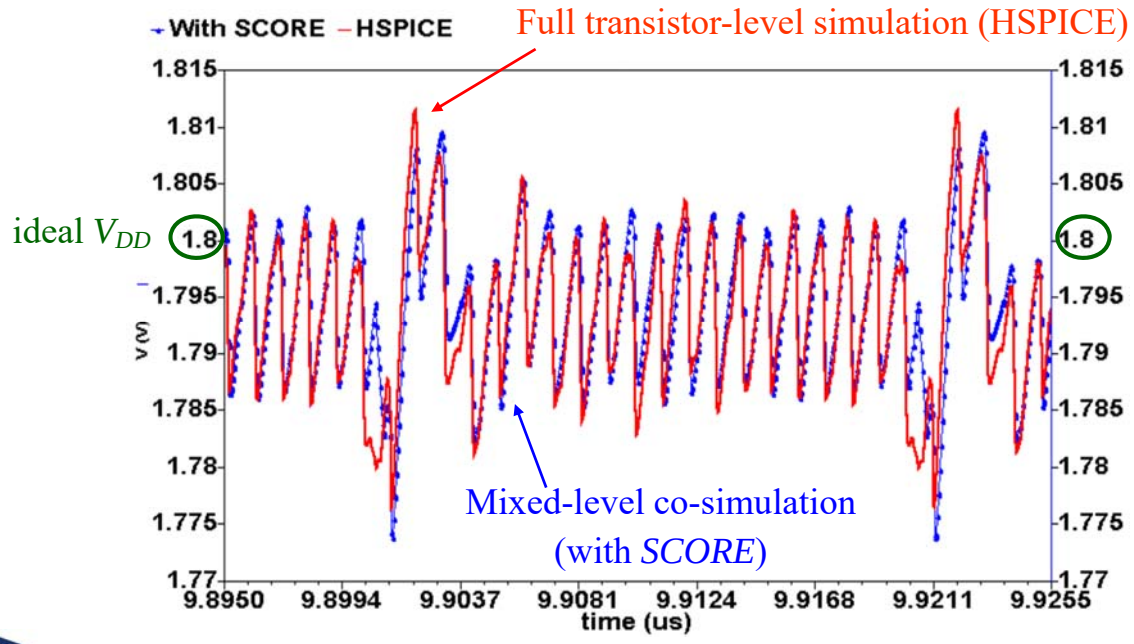
Without *SCORE* Macromodel

- Replace the PLL nelist by
  - **Only using a noise-aware PLL behavioral model**



Recursive Approach

- Replace the PLL nelist by
  - **PLL behavioral model + *SCORE* macromodel**



Comparison Results

Ideal V_{DD} = 1.8V		With supply noise interaction			No interaction	
		HSPICE	With *SCORE*	|Error| (%)	No *SCORE*	|Error| (%)
Supply noise waveform	PPV (V)	1.8113	1.8095	0.1	1.8035	0.4
	NPV (V)	1.7763	1.7737	0.2	1.7881	0.7
	RMS value (mV)	6.75	6.38	5.4	5.04	25.4
	Correlation coefficient	1	0.92	-	0.72	-
PLL responses	V_{lock} (V)	0.9945	0.9943	0.02	0.9940	0.05
	T_{lock} (us)	3.505	3.482	0.7	3.561	1.6
	pk-pk jitter (ps)	34.3	35.2	2.6	27.5	19.8
$T_{extract}$ for *SCORE* (sec)		-	247	-	-	-
T_{sim} (sec)		37721	1218	-	1153	-

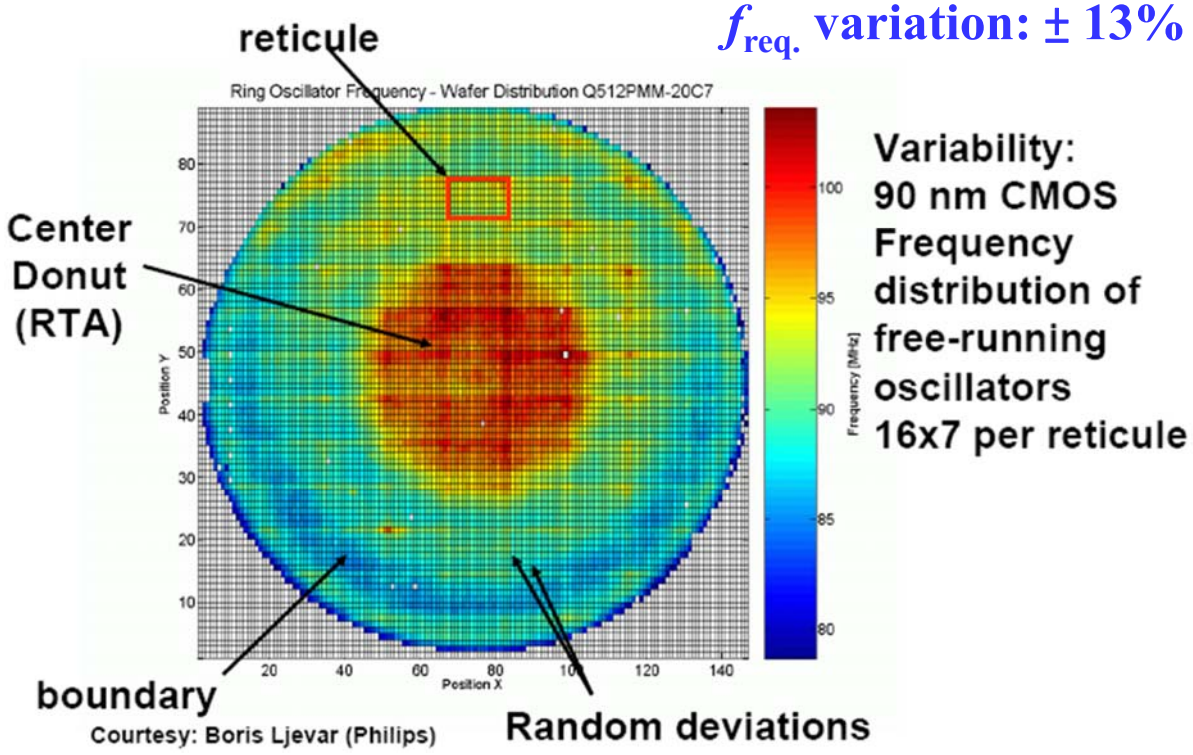
fast and accurate

Outline

- AMS circuits in SOC
- Behavioral modeling for analog circuits
- Applications of analog models
  - Noise interactions in AMS systems
    - Supply noise aware behavioral modeling
    - *SCORE* macromodel
  - Yield Enhancement
    - Analysis of process variation effects
    - Process variation aware behavioral modeling

Process Variation

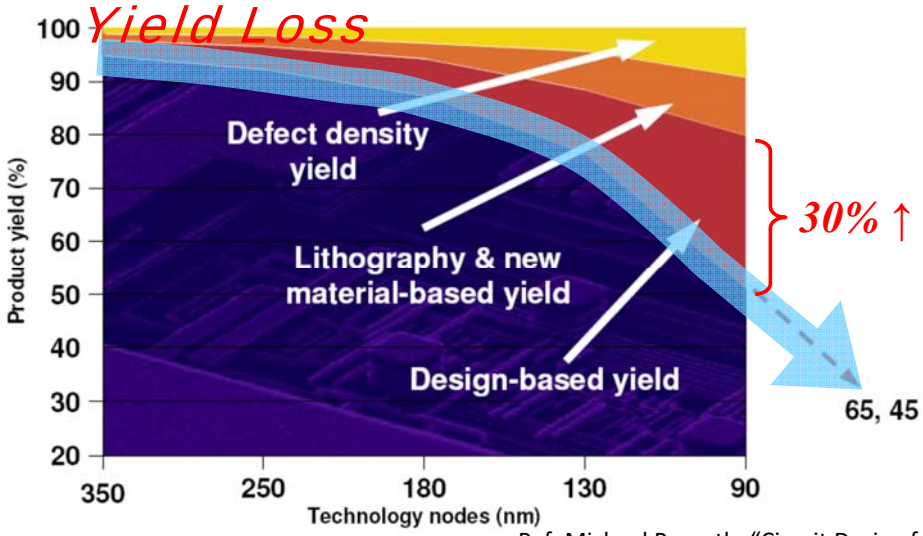
$f_{req.}$ variation: ± 13%



Courtesy: Boris Ljevar (Philips)

Yield Loss Issues

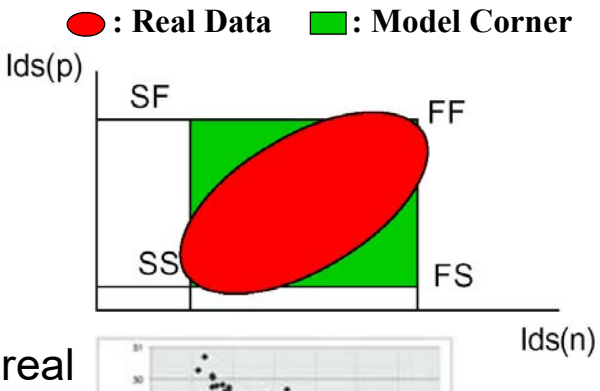
- In deep-submicron technology, yield loss issues are more and more serious
- Parametric variability will dominate yield loss

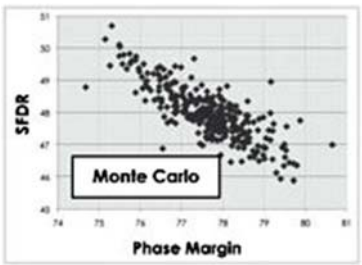


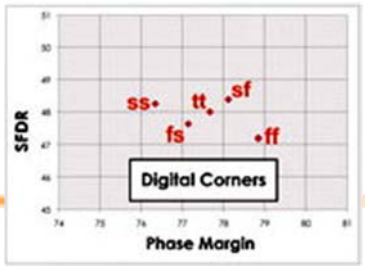
Ref: Michael Pronath, "Circuit Design for Yield with MunEDA WiCkeD", MunEDA Technical Forum 2008

Corner Simulation

- Typical verification approach: simulating the **process corners**
  - Provided in technology file
  - Require only a few simulations
- Often results in over-design
  - Things may not go so worse
- The process corner may not the real distribution corner
  - Wrong estimation
- There are more and more corners in advanced process
  - Combinations of every PVT corners (process, voltage, temperature) are huge
  - How to simulate hundreds of corners?



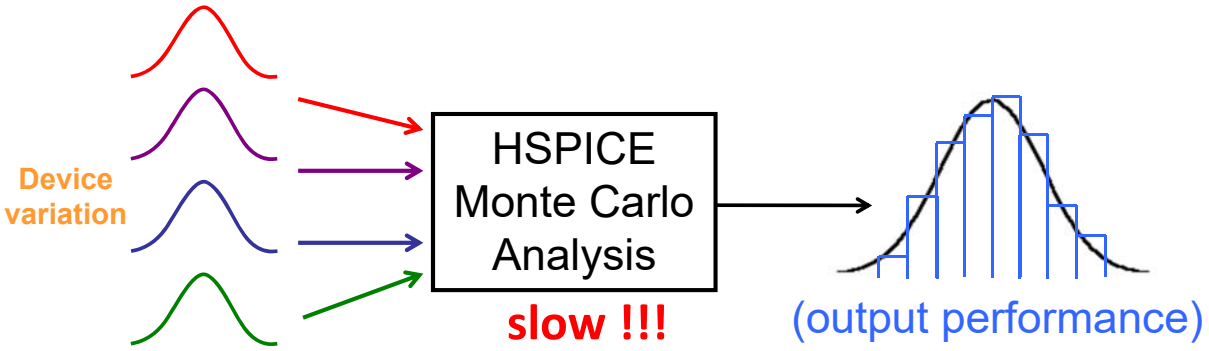




Courtesy: Solido Design Automation

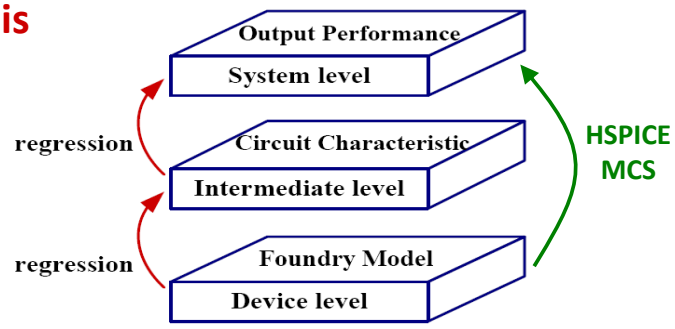
Monte Carlo (MC) Analysis

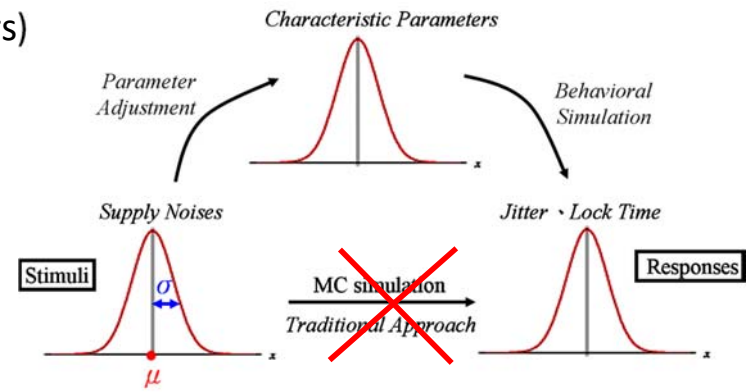
- MC analysis is a comprehensive method to check the process variation effects
  - Corner simulations often result in over design
- Simulate a circuit with **random samples of devices variations** (ex: W, L, V_t, and T_{ox} variations)
  - **Huge amount of simulations**
- **Transistor-level simulation** is often required for analog circuits
  - **Extremely time-consuming**



Hierarchical Statistical Analysis

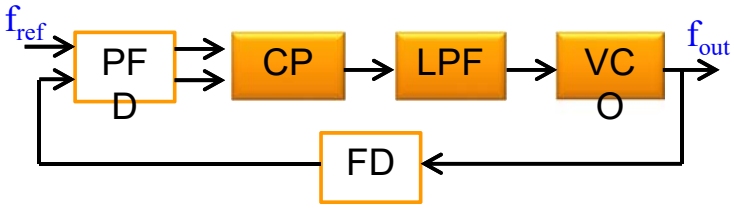
- **Hierarchical statistical analysis** is popular to improve the analysis speed
- **Regression-based approach**
  - **Regression cost** is often expensive
  - **Poor observability** (only statistical numbers)
- Our approach: use **behavioral models**
  - Avoid fitting the final performance directly





Behavioral Monte Carlo Analysis

- Behavioral Monte Carlo Simulation (BMCS)
  - Take a CPPLL design as the first study case to analyze its process variation effects



- The behavioral model accuracy is the most critical issue
  - Ideal top-down model is not accurate enough
  - Use bottom-up method to extract actual circuit properties to improve the accuracy
- Sensitivity analysis (SA) and quasi-SA models are adopted instead of RSM
  - Timing: traditional SA (linear)
  - Analog: quasi-SA (non-linear)

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 Trans. on Circuits and Systems I, pp.1160-1172, Jun. 2009.*

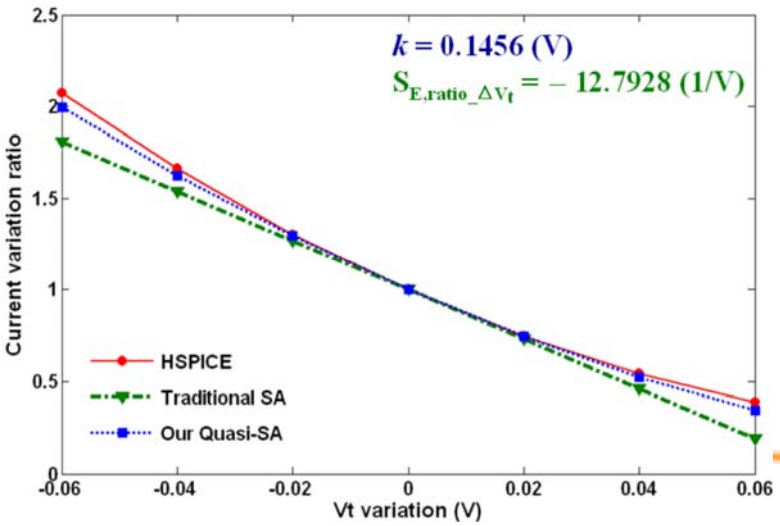
Quasi-SA for CP

- Estimate the **current variation ratio** under V_t variation:

$$ratio(\Delta V_t) = \frac{I_{CP}(\Delta V_t)}{I_{CP,0}} \cong \frac{[V_{GS} - (V_{t0} + \Delta V_t)]^2}{(V_{GS} - V_{t0})^2} = \left(\frac{(V_{GS} - V_{t0}) - \Delta V_t}{V_{GS} - V_{t0}}\right)^2$$

$$= \left(1 - \frac{\Delta V_t}{V_{GS} - V_{t0}}\right)^2 = \left(1 - \frac{\Delta V_t}{k}\right)^2$$

Extract the ***k*** value

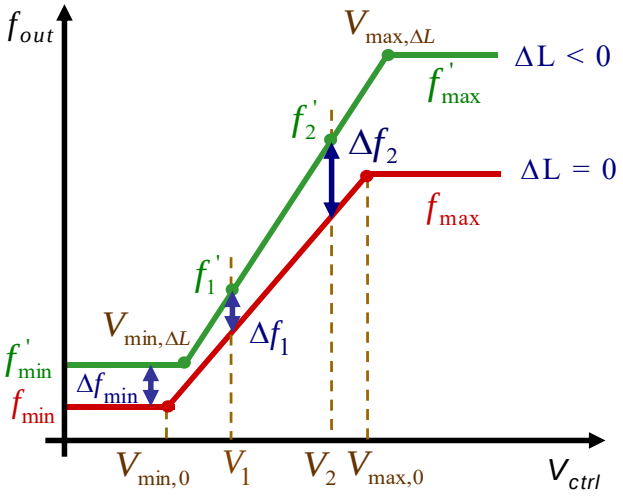


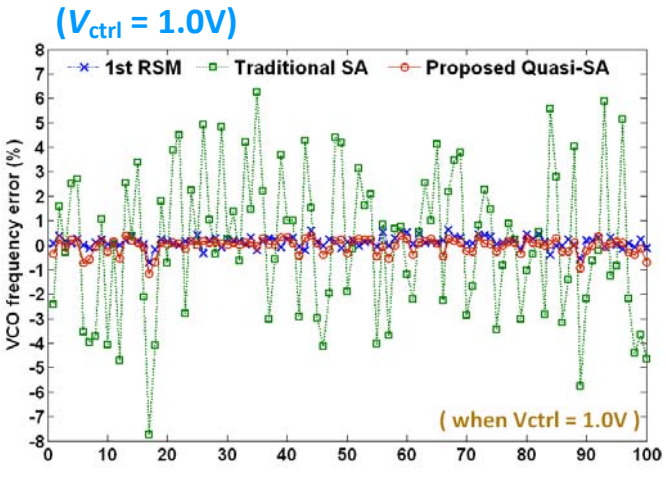
- Same extraction time as in traditional SA
- More accurate: similar to HSPICE results

Quasi-SA for VCO

- Traditional sensitivity analysis is not accurate in VCO
  - Different V_{ctrl} has different sensitivity
- Consider V_{ctrl} effects in the proposed model

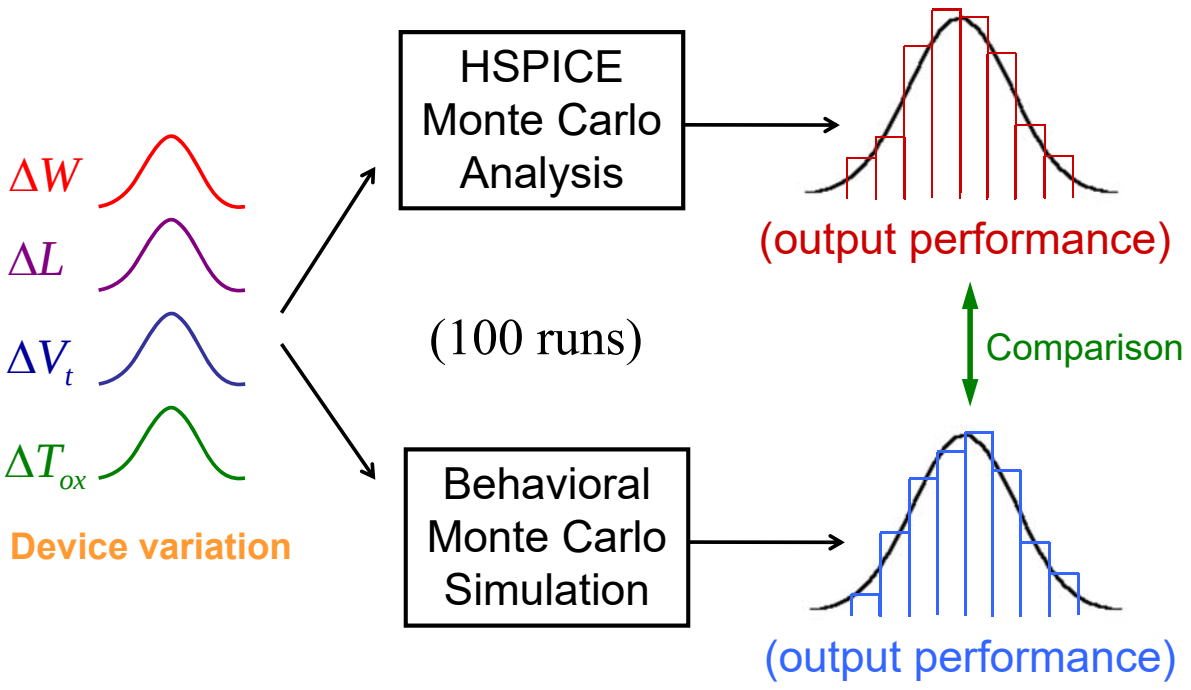
Quasi-SA model:
f_{VCO} error < ±1%





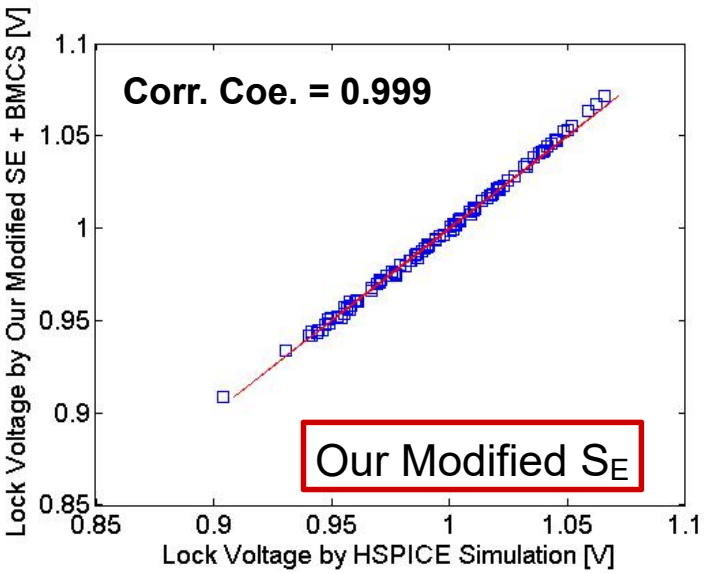
*RSM: Response Surface Methodology

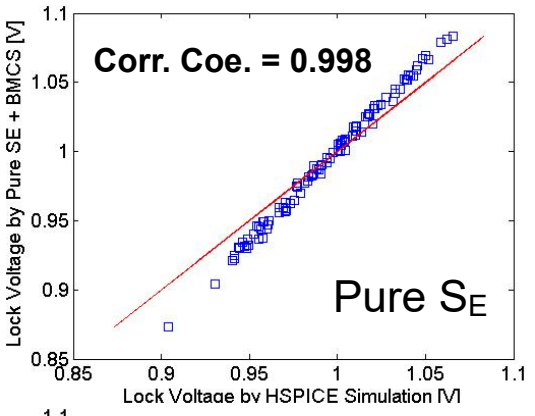
Comparison of Different MC Approaches

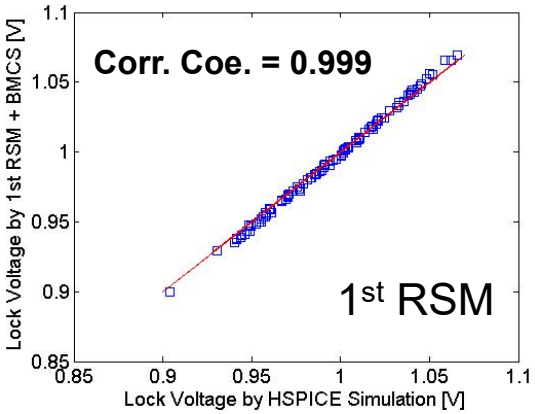


Scatter Plots – V_{lock} (100 runs)

- Corr. Coe. : correlation coefficient







Comparison Results: 100-run MCS

f_{out}: 800MHz		1st RSM + BMCS	SA + BMCS	Quasi-SA + BMCS	HSPICE
V_{lock} (V)	Mean	0.993 (-0.1%)	0.993 (-0.1 %)	**0.995 (0.1 %)**	**0.994**
	St. Dev.	0.036 (6.4%)	0.045 (32.1 %)	**0.035 (1.7 %)**	**0.034**
T_{lock} (us)	Mean	3.449 (2.2%)	3.441 (2.0 %)	**3.438 (1.9 %)**	**3.374**
	St. Dev.	0.573 (-0.6%)	0.541 (-6.2 %)	**0.572 (-0.8 %)**	**0.576**
pk-pk Jitter (ps)	Mean	12.2 (-7.6%)	12.4 (-6.1 %)	**12.4 (-6.1 %)**	**13.2**
	St. Dev.	1.36 (-2.9%)	2.29 (63.6 %)	**1.41 (0.7 %)**	**1.40**
	Worst	16.6 (-2.4 %)	16.4 (-3.5 %)	**16.7 (-1.8 %)**	**17.0**
$T_{extract}$ (hr)		34.2	8.55	**8.55**	**N/A**
$T_{sim.}$ (hr)		2.95	2.93	**3.50**	**598.54**

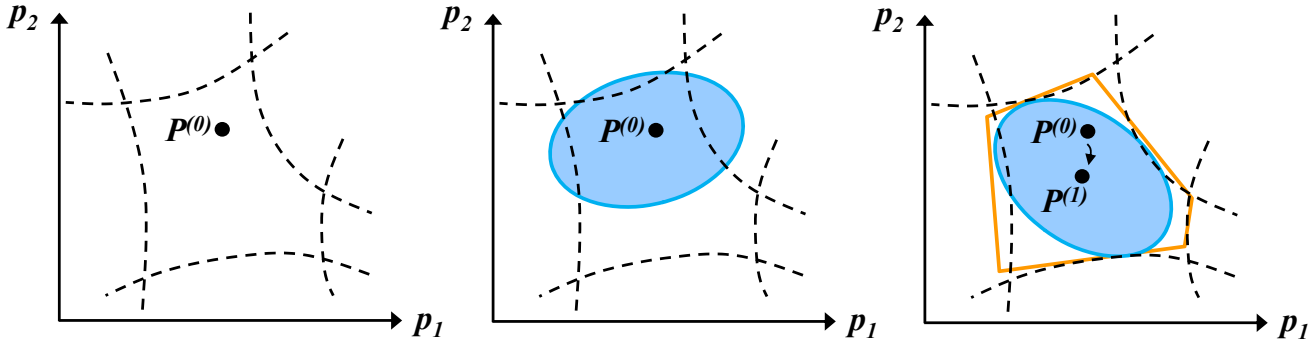
St. Dev. : Standard Deviation

Low regression cost & accurate

Design Centering

- In early design stages, process variation impacts must be considered to reduce the yield loss
- **Design centering** approach is often used to improve the design-based yield
- Need to extract the boundary of the feasible design region before computing the "center" → **a heavy**

boundary extraction ➡ **yield analysis** ➡ **design centering**

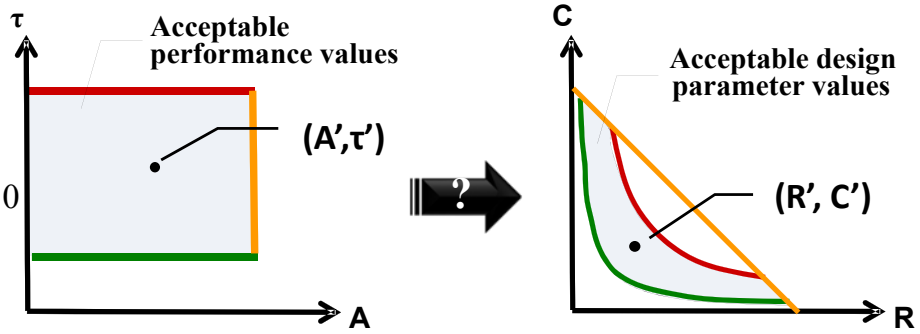


Acceptable Design Region

- For analog circuits, it is not easy to figure out the acceptable design region at device level
  - Ex: a simple RC low-pass filter

$A = R + C \le 4.0$

$0.5 \le \tau = RC \le 2.0$

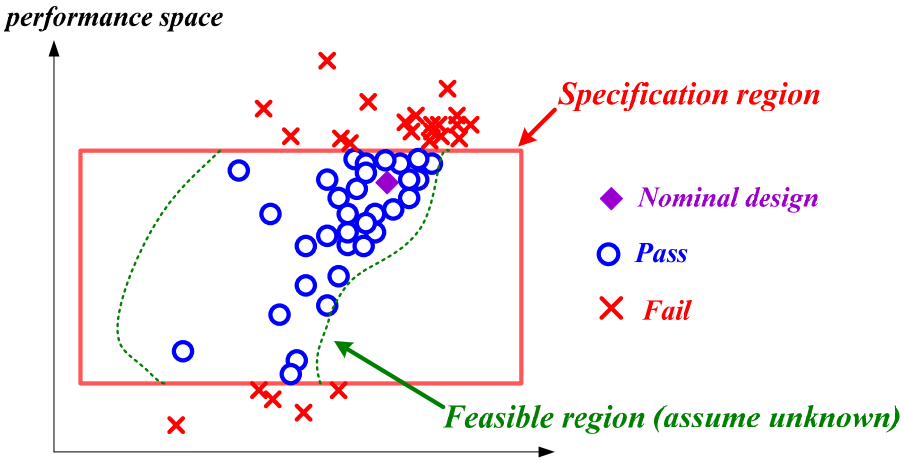


- Numerous design constraints are also big troubles
  - Ex: optimize a PLL using geometric programming approach
  - 40,000 optimization variables
  - 150,000 design constraints

Ref: Helmut E. Graeb, "Analog design centering and sizing", Springer, 2007.
D. Colleran et al. "Optimization of Phase-Locked Loop Circuits via Geometric Programming," CICC, 2003.

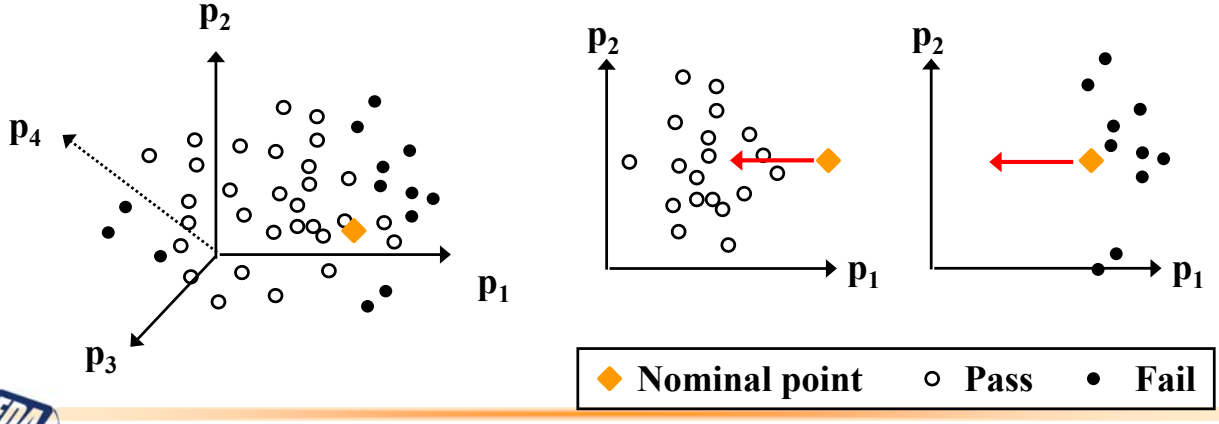
Why not Forgetting the Boundary ?

- Performance distribution in the yield analysis has partial information of the feasible region
- **Reusing yield analysis results** has no extra simulation cost
  - Yield analysis is essential in the design flow
  - The heavy overhead for boundary extraction can be avoided



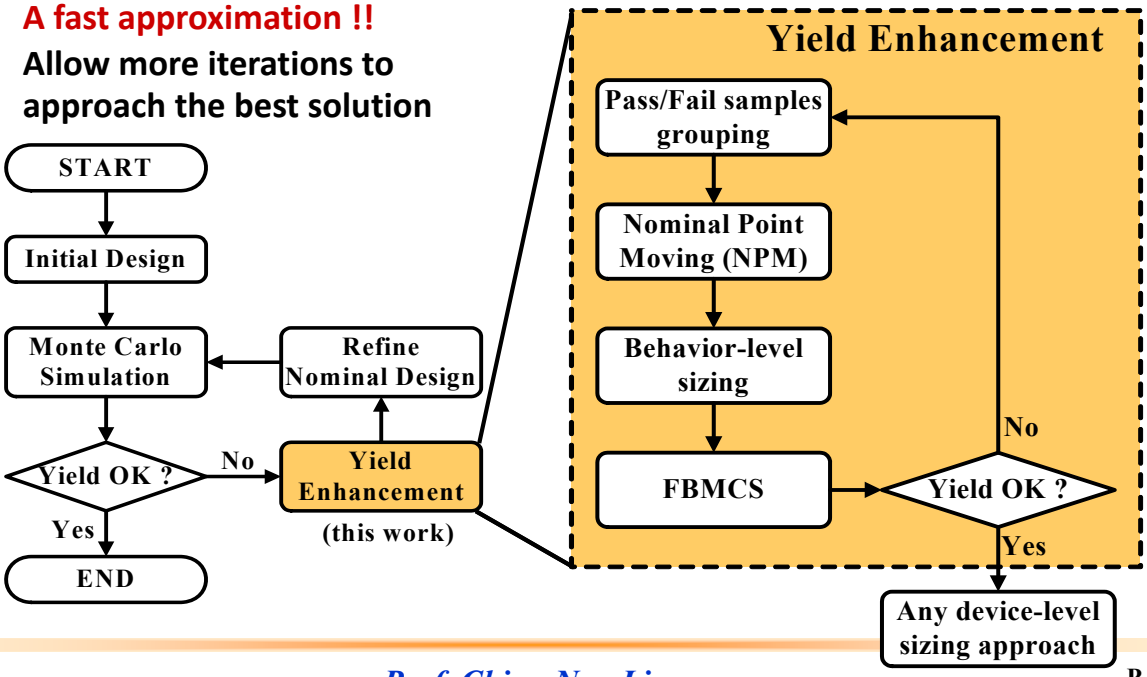
Force-Directed Nominal Point Moving

- Force-directed **nominal point moving (NPM)** algorithm
  - Close to the Pass group (attraction)
  - Far from the Fail group (repulsion)
  - The force equilibrium point is the new nominal point
- Iteratively calculate the forces until the nominal point is stable
  - May not find the best location in one calculation
  - Push the nominal point to better location gradually



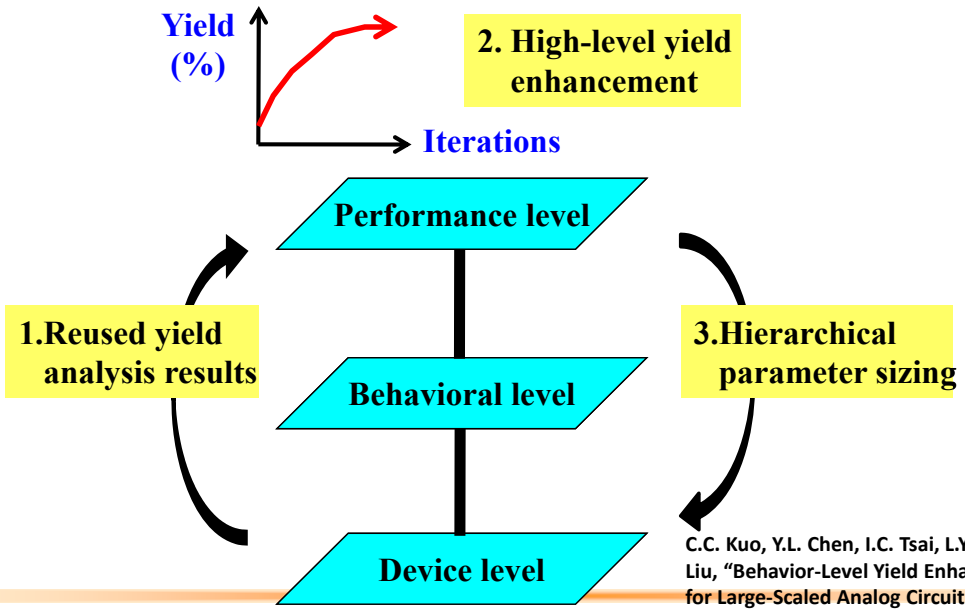
Proposed Yield Enhancement Flow

- Three major steps in the iterative yield enhancement flow
  - Use force-directed model to avoid boundary extraction cost
- Behavior-level yield enhancement approach is much faster
  - A fast approximation !!
  - Allow more iterations to approach the best solution



Behavior-Level Yield Enhancement

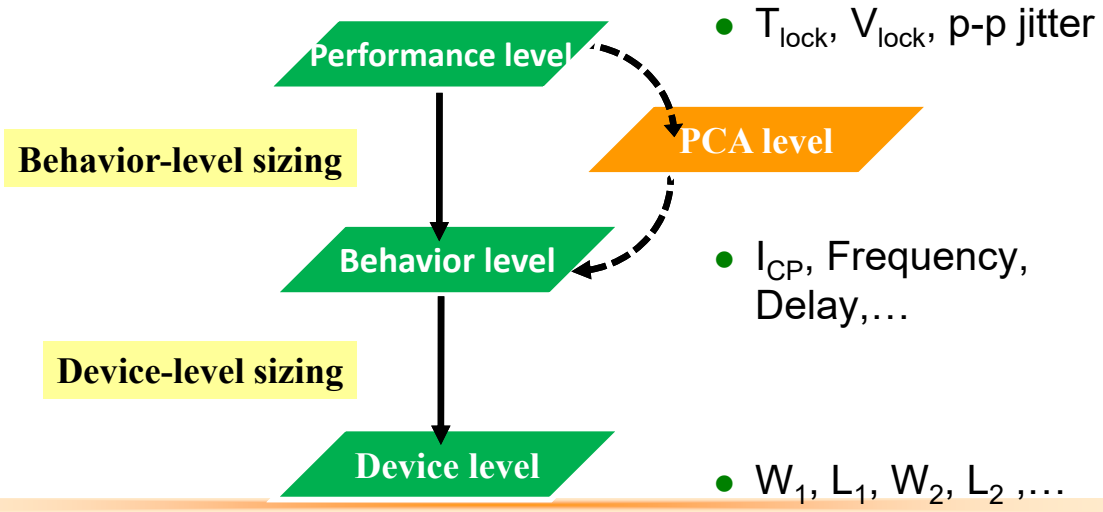
- Perform all steps at performance level and behavior level
  - Reduce the iteration time of the sizing-evaluation loop
- Perform device-level sizing/evaluation after yield enhancement
  - Become a one-time cost → shorten yield enhancement process



C.C. Kuo, Y.L. Chen, I.C. Tsai, L.Y. Chan, and C.N. Liu, "Behavior-Level Yield Enhancement Approach for Large-Scaled Analog Circuits", DAC 2010.

Behavior-Level Sizing

- **Map the NPM results into behavioral parameters**
- Hierarchical approach is often used for complicated analog circuits, such as PLLs
- **Add an extra PCA level** to reduce the regression efforts
  - PCA = Principal Component Analysis

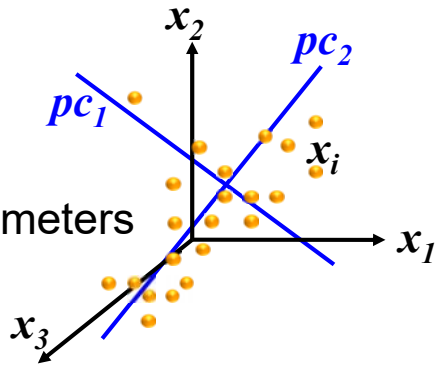


Principal Component Analysis

- Find a few linear combinations of behavioral parameters (B_j) to represent the numerous samples
  - PCs are orthogonal to each other

$$PC_i = \sum_{i<j,j=1}^{n} p_{ij} B_j$$

- **Dimension reduction**
  - Often results in fewer variables
- Include the **correlation** between parameters
  - Avoid infeasible solutions
- Example:
  - Variables: x_1,x_2,x_3
  - Principal component: $pc_1(x_1,x_2,x_3)$,pc_2 (x_1,x_2,x_3)

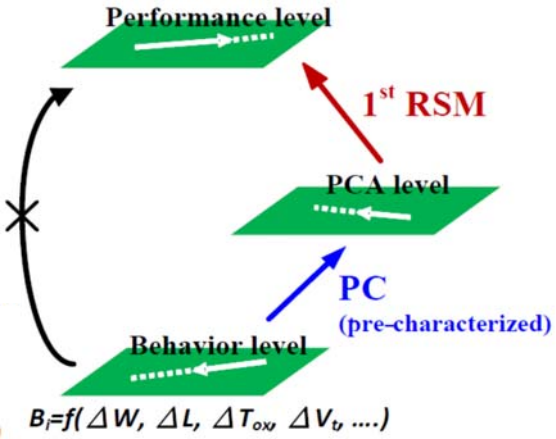


$$pc_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots$$
$$pc_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots$$
$$\vdots$$

FBMCS

- Is the yield of the new nominal design satisfied?
- How to perform next NPM without new distribution?
  - ⇨ Fast Behavioral-level Monte Carlo Simulation (FBMCS)
- It's an equation-based MC simulation
  - Generate random samples at behavior level
  - Translate to performance distribution through RSM equations
- Provide rough yield analysis to **guide next NPM**
  - Shorten the iteration time
  - Accurate analysis can be done after the iterations

PCs to performance: $S_i = e_{io} + \sum_{j=0}^{m-1} d_{ij} PC_j$



Case Study: CPPLL

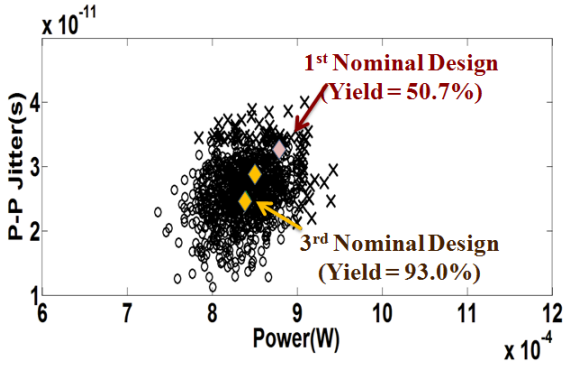
- This Charge-Pump PLL is consisted of 5 blocks
  - Phase-frequency detector (PFD), voltage-controlled oscillator (VCO), charge-pump (CP), low-pass filter (LPF), and frequency divider (FD)
  - Contain 163 transistors
- TSMC RF 0.18μm process
  - Process variation : ΔW, ΔL, ΔVt, ΔTox
- Simulation environment MC simulation:
  - 1000-run Behavioral MC simulation [17]
  - Analog Artist (Cadence) + Spectre
- Yield enhancement calculation :
  - Stop iterations when the yield value is stable within 1%
- Device sizing:
  - WiCkeD (MunEDA)

Performance	Specification
Lock voltage	1 V ± 0.2V
Lock time	< 5us
P-P jitter	< 34ps
Power	< 0.9mW

[17] 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 Trans. on Circuits and Systems I*, Jun. 2009.

Nominal Point Moving Results

- Start from an existing design
  - Nominal point meets all spec.
  - Initial yield is only 50.7%
- Yield: 50.7% → 93.0%
  - Only need 3 iterations
- Total run time: 4.43 seconds



Blocks	Parameters	Spec.	Initial Design	After Enhance
CPPLL (nominal)	V_{lock} (V)	1 ± 0.2	0.99	**1.08**
	T_{lock} (μs)	< 5	1.17	**1.50**
	pk-pk jitter (ps)	< 34	32.72	**28.77**
	power (mW)	<0.9	0.88	**0. 85**
CP	Current ratio I_{ratio}	—	1.00	**1.00**
	Switch time T_{sw} (ns)		4.64	**5.01**
VCO	K_{VCO} (GHz/V)		1.14	**1.00**
	f_{min} (MHz)		317.3	**275.5**
	f_{max} (MHz)		1036.1	**904.1**
Yield (%)			50.7	**93.0**

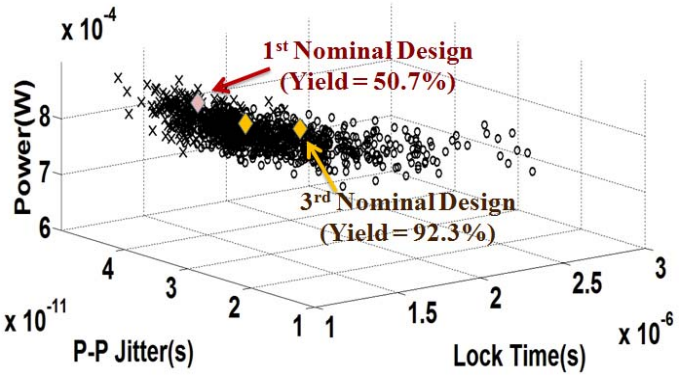
Device-Level Sizing Results

- Use commercial sizing tool to obtain the corresponding device sizes of the final nominal point
- Block-level sizing improves the sizing speed
  - Use the obtained behavioral parameters as the sizing target of each sub-block
  - Device sizing is performed block-by-block → faster
  - Use the spec. directly for sizing the whole circuit is too slow
- A new circuit with the expected yield is really achievable !!

Blocks	Parameters	Spec.	Behavioral results	Whole PLL sizing	Block-level sizing
CPPLL (nominal)	V_{lock} (V)	1 ± 0.2	1.08	**1.11**	**1.09**
	T_{lock} (μs)	< 5	1.50	**1.39**	**1.33**
	jitter (ps)	< 34	28.77	**25.59**	**26.9**
	power (mW)	<0.9	0.85	**0.86**	**0.86**
Yield (%)			93.0	**91.5**	**92.3**
Area (um^2)			--	**416.45**	**417.30**
Optimization Time (min.)			0.07	**300**	**6**

Run Time Comparison

- Previous work about CPPLL circuits [3]
  - Use performance space exploration (PSE) method to extract the part of acceptable design region → 4-5 hours
  - Behavior-level sizing → 1-2 hours
- Only need 4.43 seconds by using the proposed approach
  - Including NPM, behavior-level sizing, and FBMCS
- The final result has been confirmed with device-level sizing and evaluation



[3] J. Zou, D. Mueller, H. Graeb, and U. Schlichtmann, "A CPPLL Hierarchical Optimization Methodology Considering Jitter, Power and Locking Time", DAC'06

Conclusions

- Analog behavioral modeling is indeed useful for AMS system verification
  - Top-down behavioral modeling is for new designs
  - Bottom-up behavioral modeling is suitable for IP-based SOC designs
- Accurate behavioral models also have many useful applications
  - Analyze supply noise effects
  - Analyze design yield and make improvement
- CAD could also be useful to analog designers
  - Waiting for more investment !!!