Analog Behavior in Custom IC Variation-Aware Design

(Invited Special Session Paper)

Trent McConaghy
Co-founder & CTO
Solido Design Automation Inc.

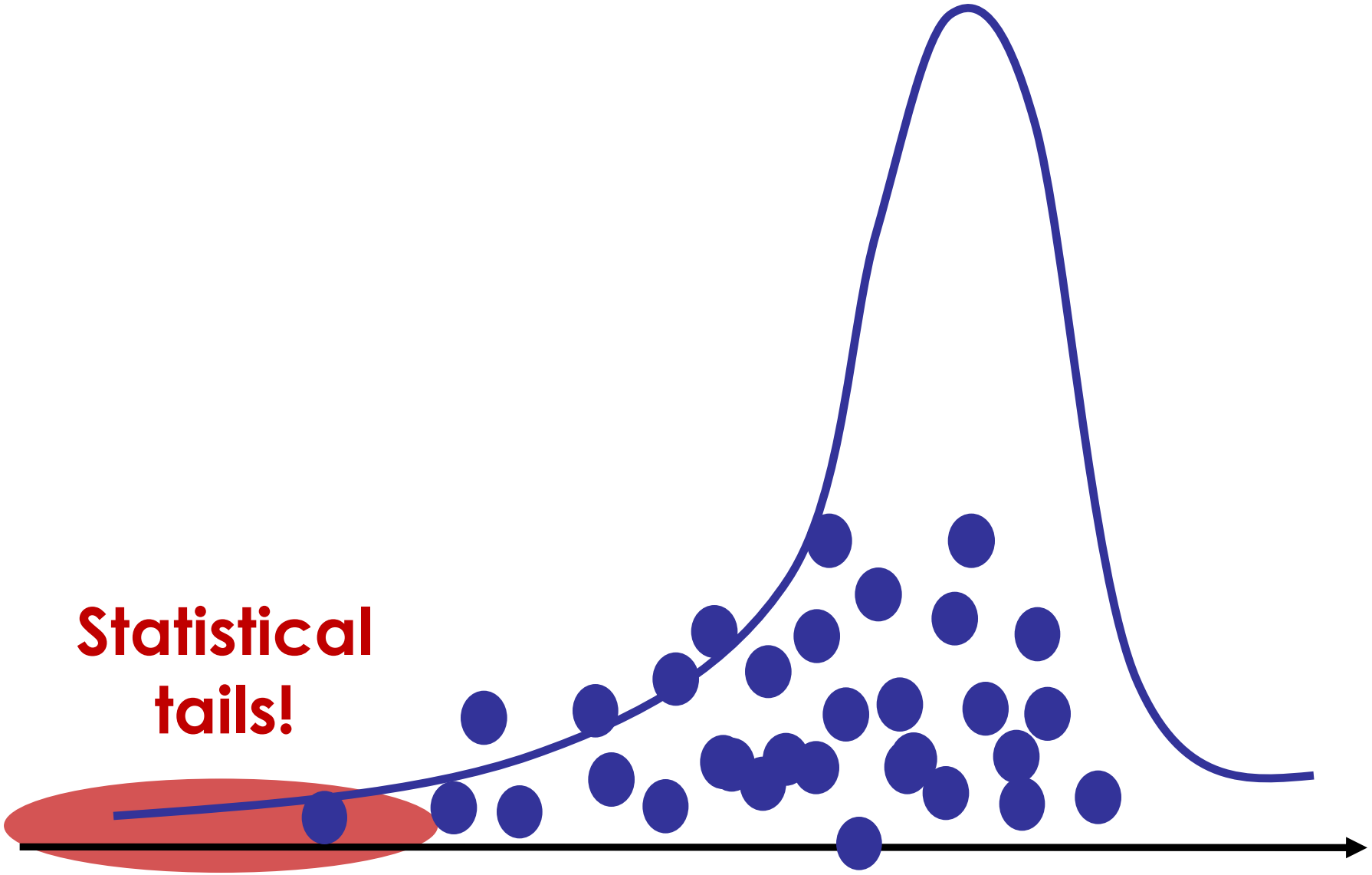
ICCAD, November 2013

solido DESIGN AUTOMATION

Tails

Tails

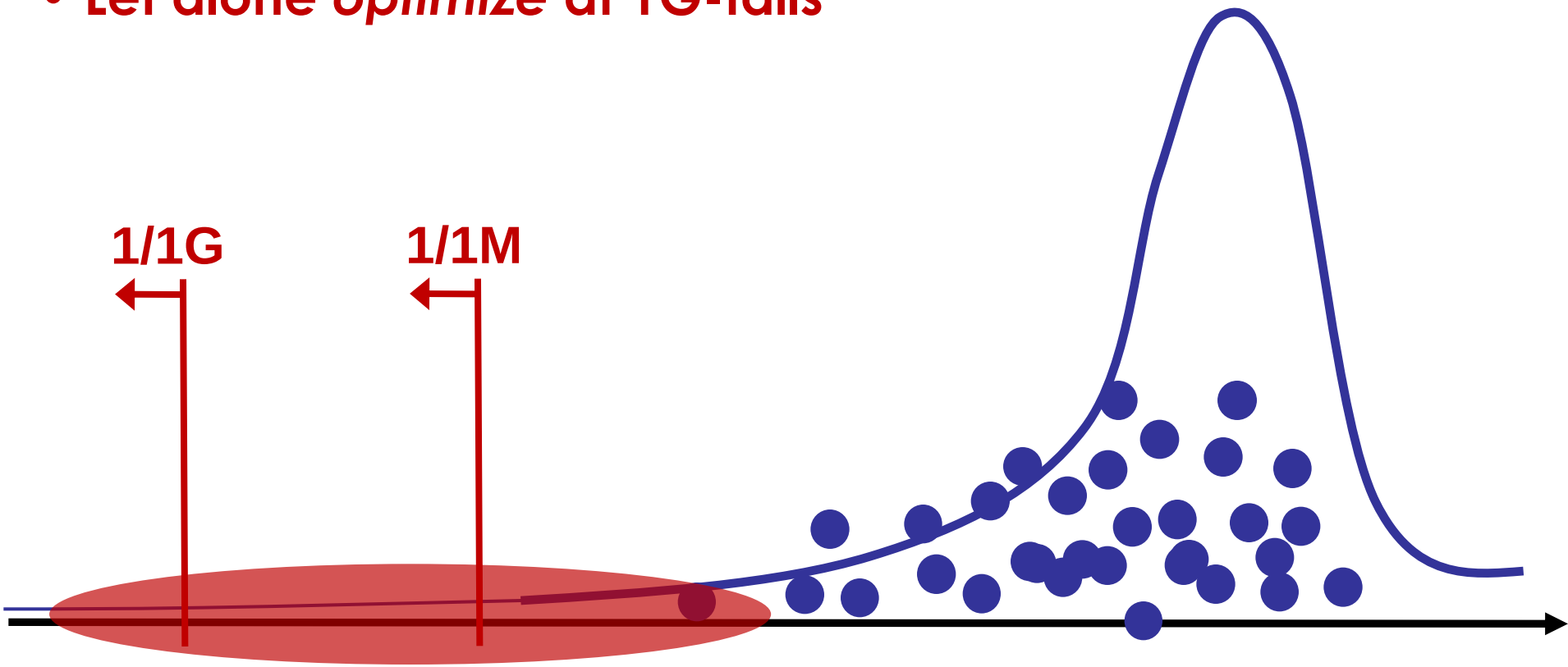
Tails



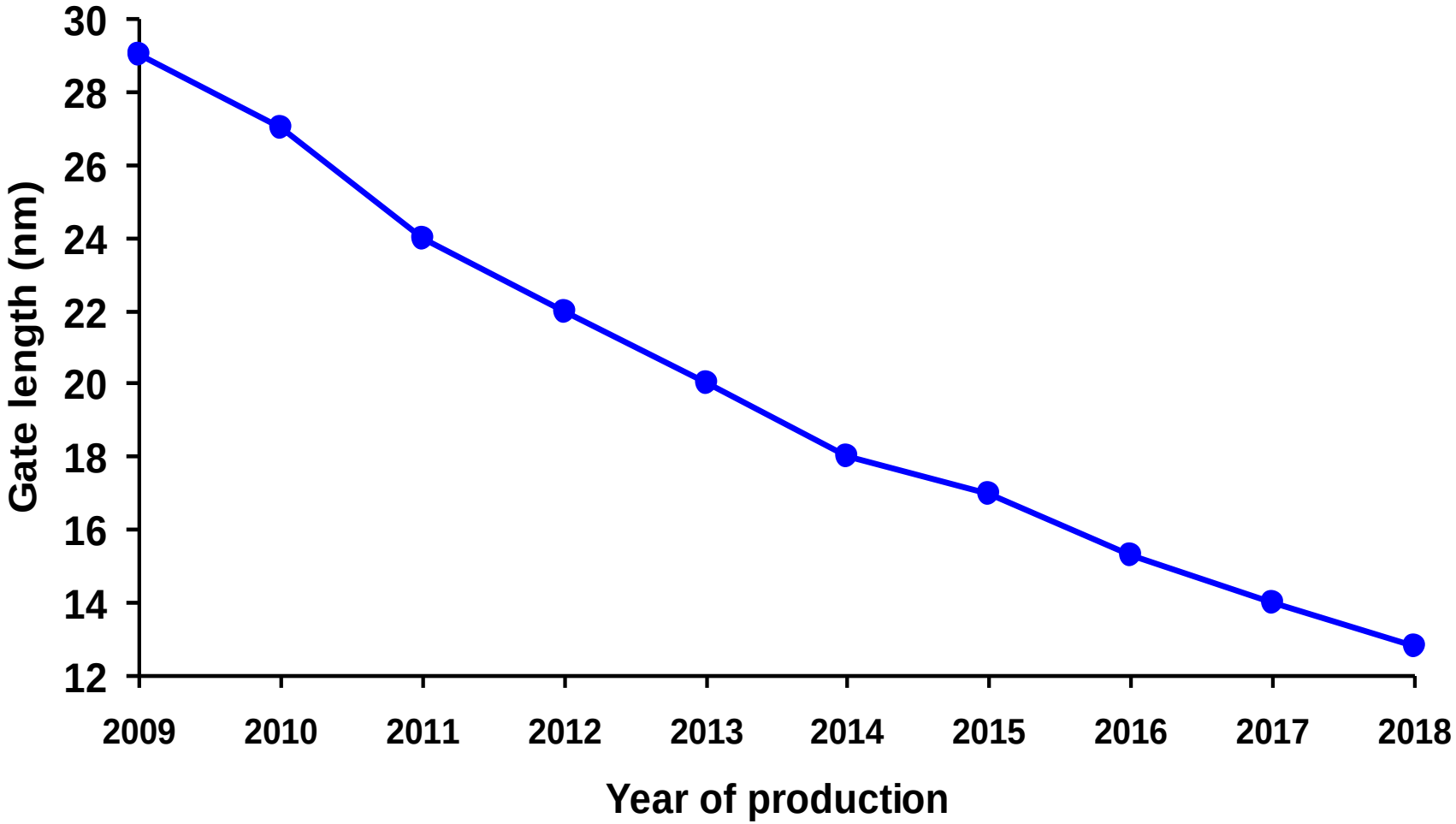
By definition, *statistical tails* are improbable.

Challenges:

- Infeasible to *simulate* 1G samples
- Let alone *optimize* at 1G-tails

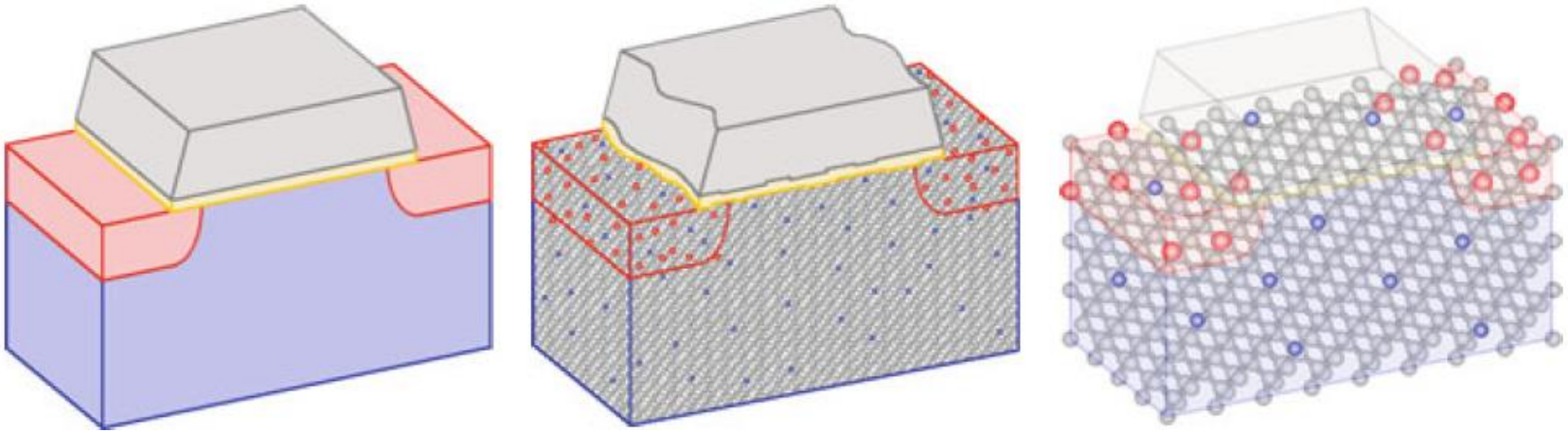


Due to Market Pressure (and History), Transistors Are Shrinking…



[ITRS, 2011]

Transistors are shrinking
But atoms aren't!



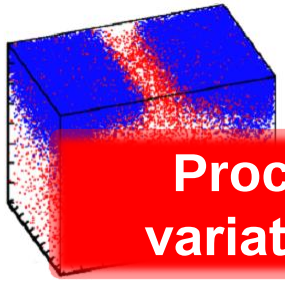
traditional

22nm

sub 10 nm (2020)

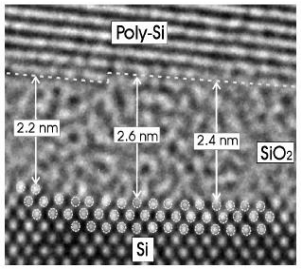
A. Asenov, "Statistical Nano CMOS Variability and Its Impact on SRAM", Chapter 3, A. Singhee and R. Rutenbar, Eds., Extreme Statistics in Nanoscale Memory Design, Springer, 2010

Variation = atoms out of place
…Propagating from devices to performance & yield



Process variation ↑

Random dopant effects

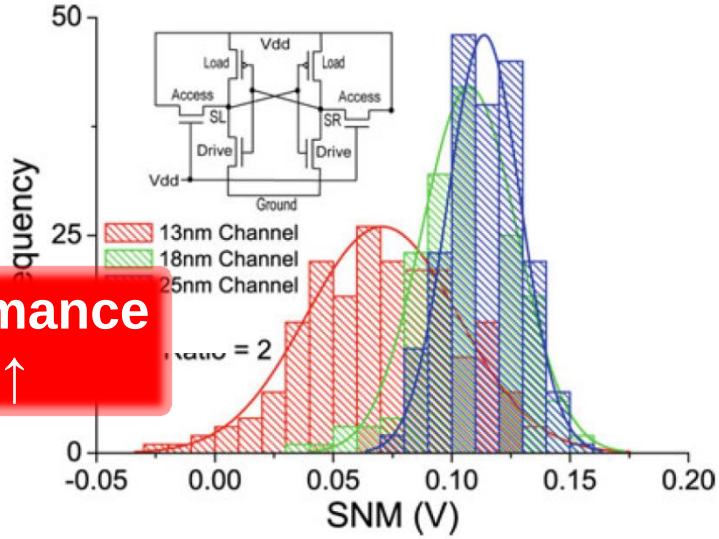


Oxide thickness

▪
▪
▪

Device performance variation ↑

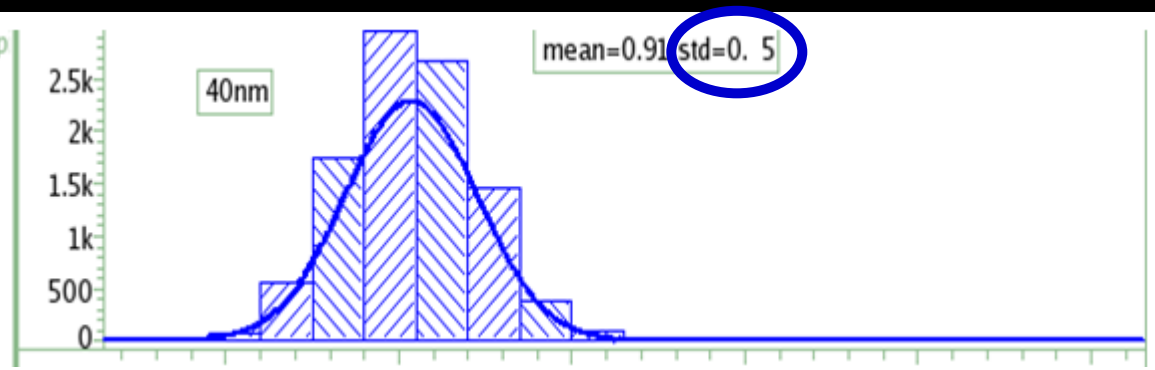
Circuit performance variation ↑



Modern Process Nodes Have High Variation … and It's Getting Worse

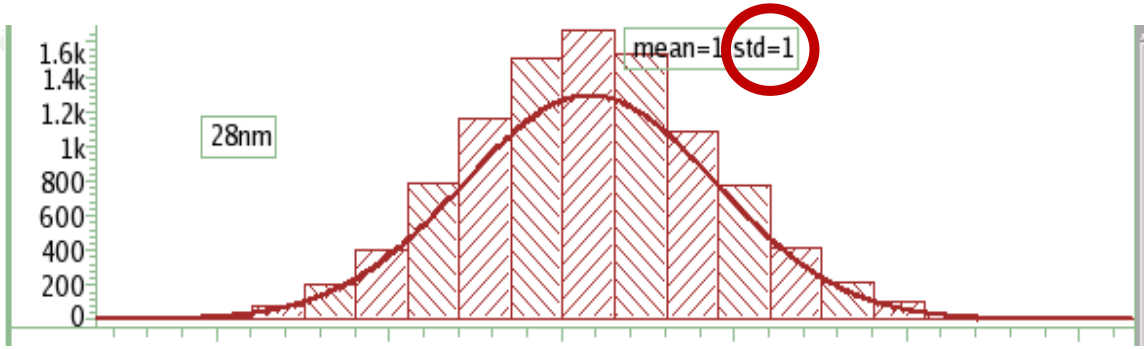
Example: GF 40-nm vs. 28-nm on Single Device

40-nm



≈2x overall increase in i_{dsat} variation from 40 to 28-nm (for global process variation)

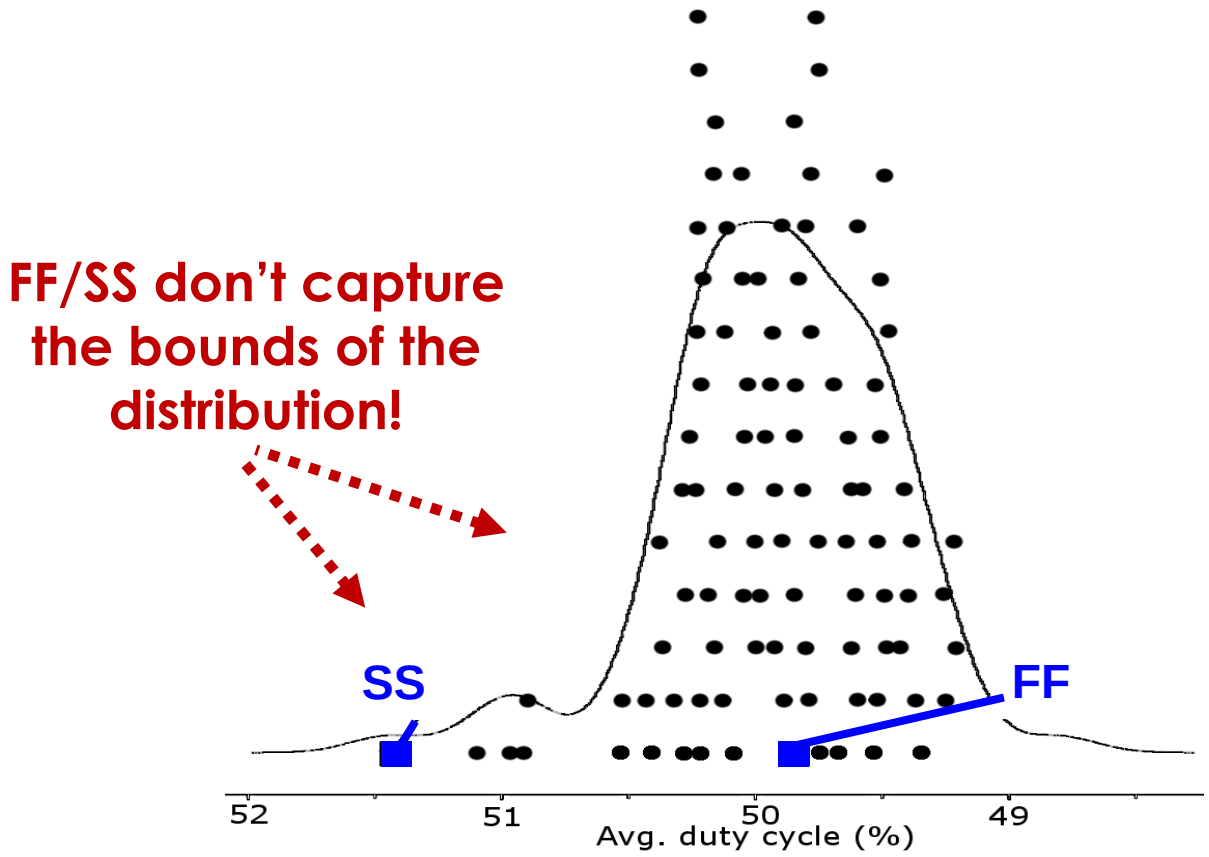
28-nm



Increased variation was also observed for mismatch variation, and for global+mismatch combined; for both nmos and pmos.

[Yao et al, DAC 2012]

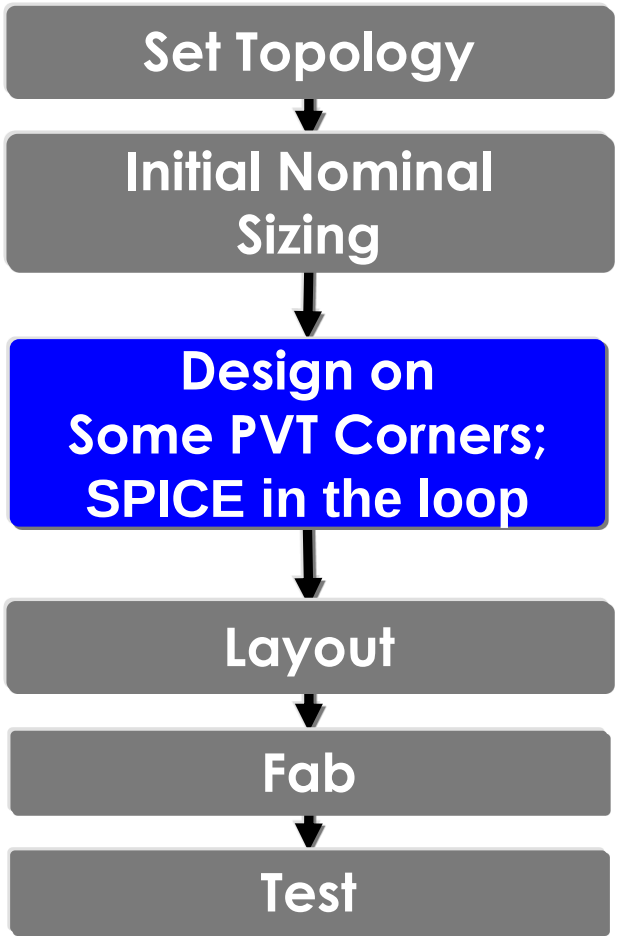
Applicability of FF/SS Corners?
Example on GF 28-nm, PLL VCO, Avg. Duty Cycle



Why: FF/SS were designed to bracket performance at the *device level*, for (digital) speed and power outputs.

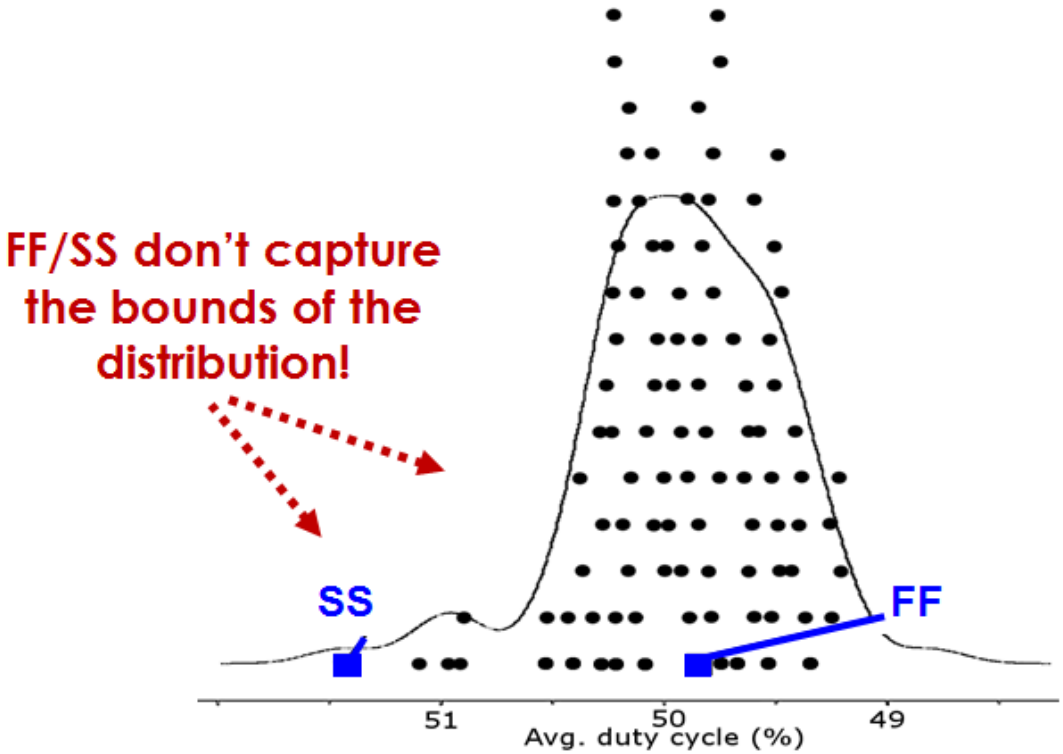
Therefore not well-suited to analog / RF circuits.

Variation-Handling Flow: PVT (Where the "P" is FF/SS corners)

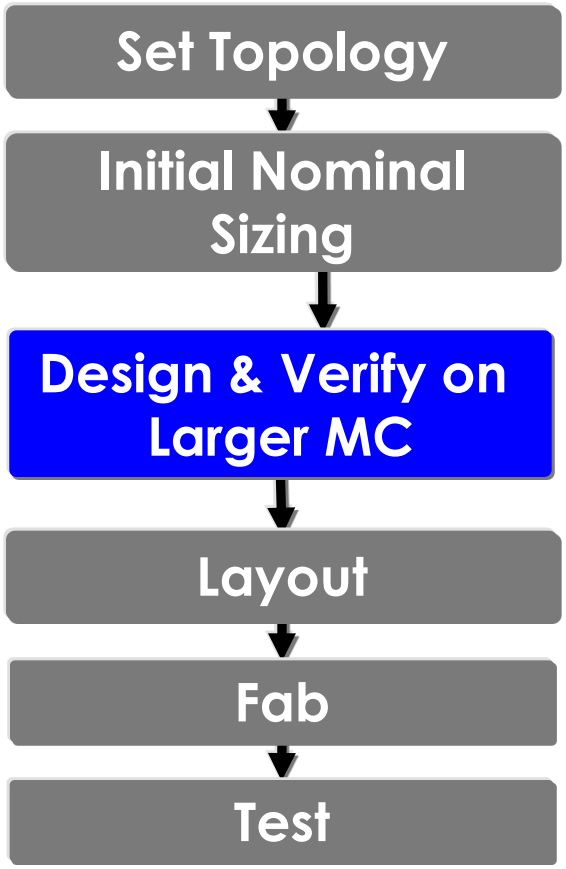


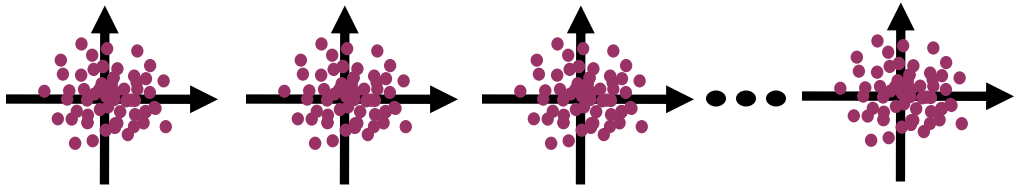
Pros: familiarity, speed, scalability

Cons: {ignores local variation, poor model of global} → hurts yield, performance



Variation-Handling Flow: Direct MC

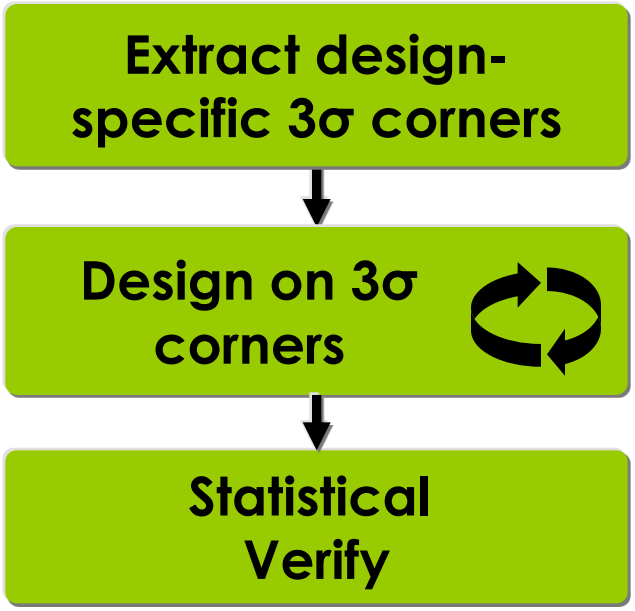


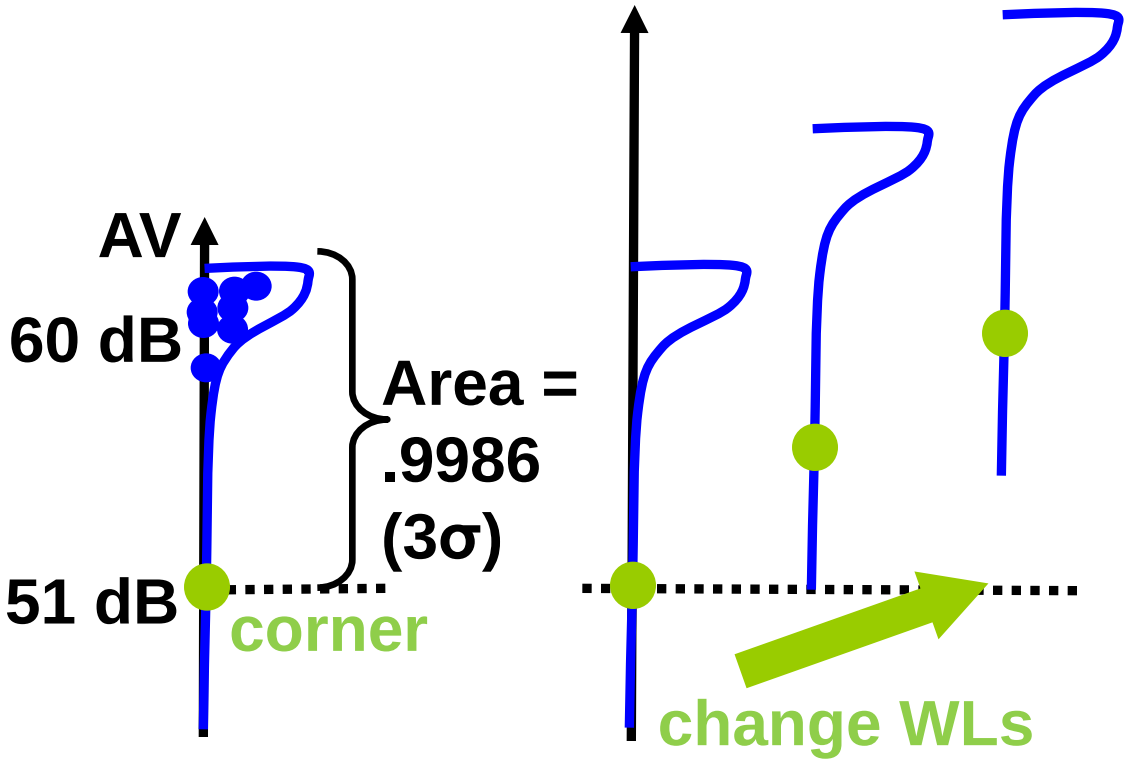


Pros: familiarity, accurate verification, scalability

Cons: *way* too many sims

Idea: Use corners. Just make the corners *good*.
This is the *sigma-driven corners flow*.





3σ Yield == Yield of 99.73% (not 3 standard deviations)

Tool support for a sigma-driven corners flow

Extract design-specific 3σ corners

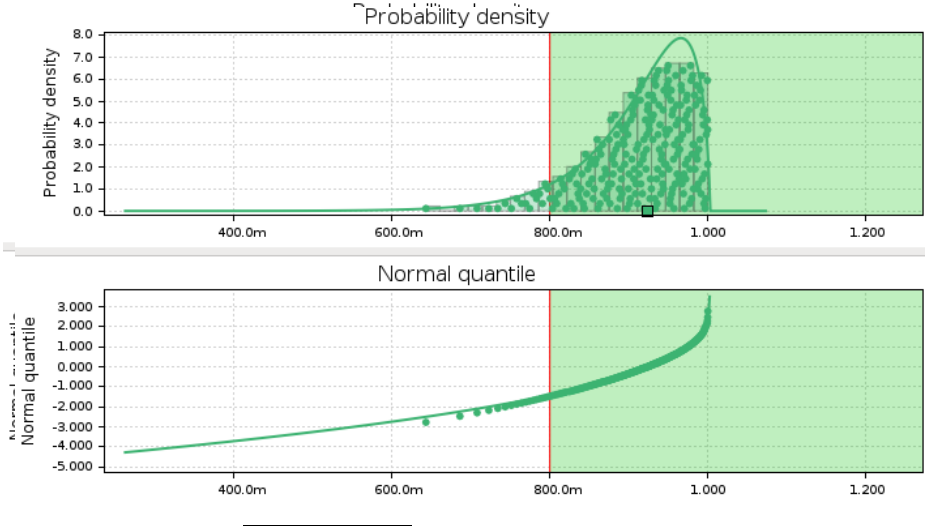
Design on 3σ corners

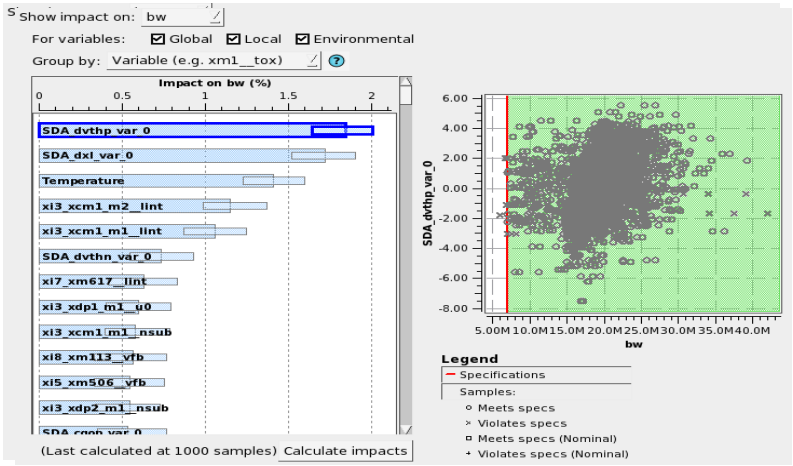
Statistical Verify

Make this *great*
e.g. Fast MC tool 3-σ CX feature

Make this *great*
e.g. Sensitivities/sweeps tool

Make this *great*
e.g. Fast MC tool fast verification feature





On "Analog"

- Analog =
  - Continuous in time, or
  - Continuous in value (voltage, current)
- **Q: Where do we need to do analog analysis?**

  (i.e. analyze continuous in time, or value)
- **A:**
  - Traditional "analog" circuits (opamps, bias generators, ..)
  - Mixed analog-digital circuits (ADCs, ..)
  - **Bitcells, sense amps**
    - **Especially when variation is present!**
  - **Standard cells**
    - **Especially when variation is present!**
- Another Q that currently has the same A: where do we need to run SPICE-based analysis?

solido DESIGN AUTOMATION

Challenges in Memory & Standard Cells

- Plain Monte Carlo is too expensive
  - Failure rate is ≈1/1M to ≈1/1G. **Tails!**
  - Running 10M to 10G+ MC samples is infeasible
- Automated sizing is common industrial practice
  - E.g. optimize a *library* of standard cells
- How to reconcile these in the *sigma-driven corners flow*?

Sigma-driven corners flow: Variant for memory & standard cells

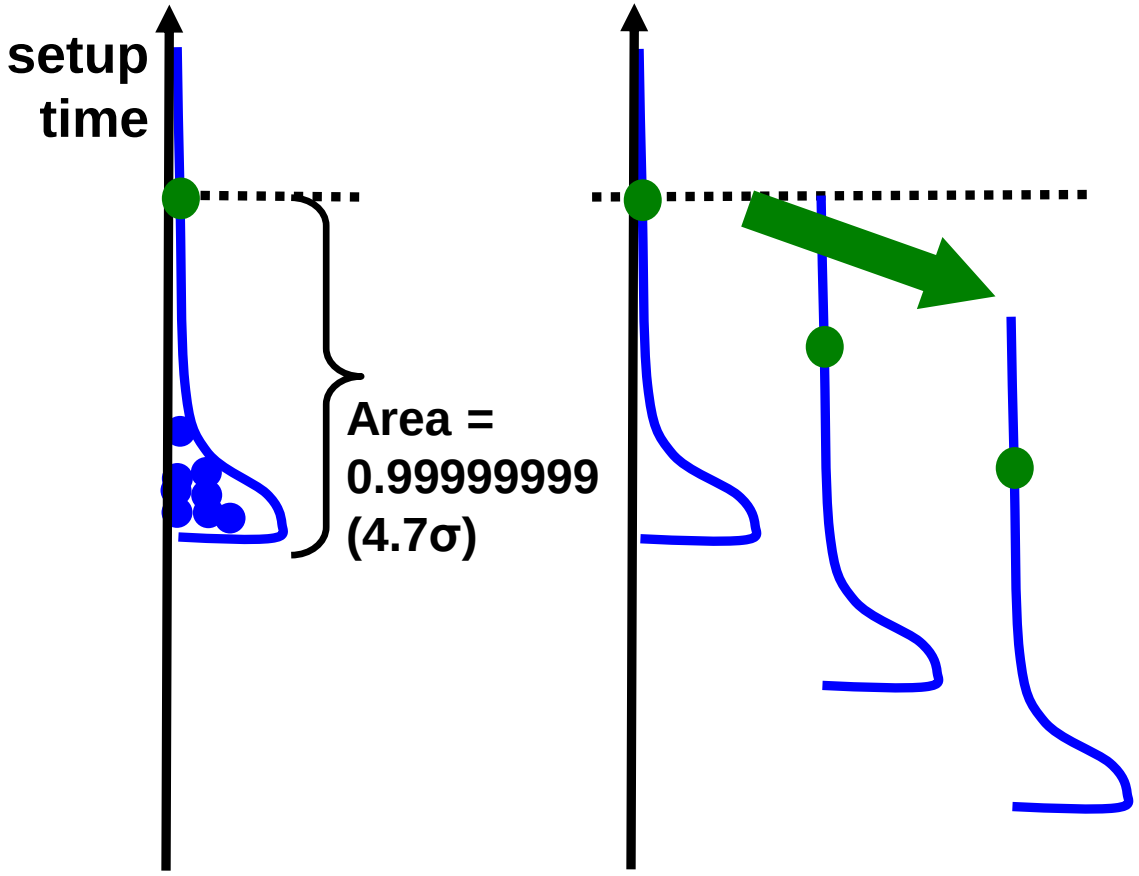
Extract design-specific *high-σ* corners

↓

Automatically size on high-σ corners

↓

High-σ statistical Verify



Tool support for a sigma-driven corners flow
For high-σ memory and standard cells

Extract design-specific *high-σ* corners

Make this *great*
e.g. High-Sigma Monte Carlo
(100x+ faster than plain MC)

↓

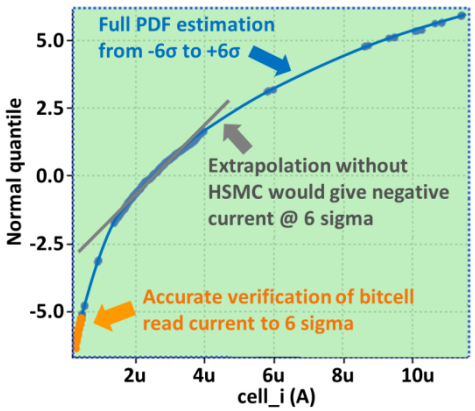
Automatically size **on high-σ corners** ↻

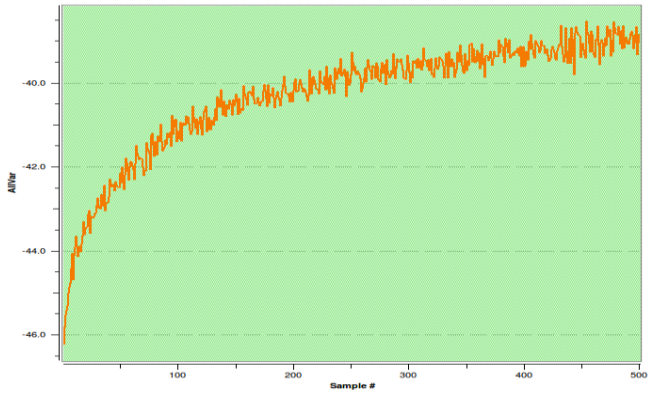
Make this *great*
e.g. optimization tool
(fast SPICE-accurate global optimization)

↓

High-σ **statistical Verify**

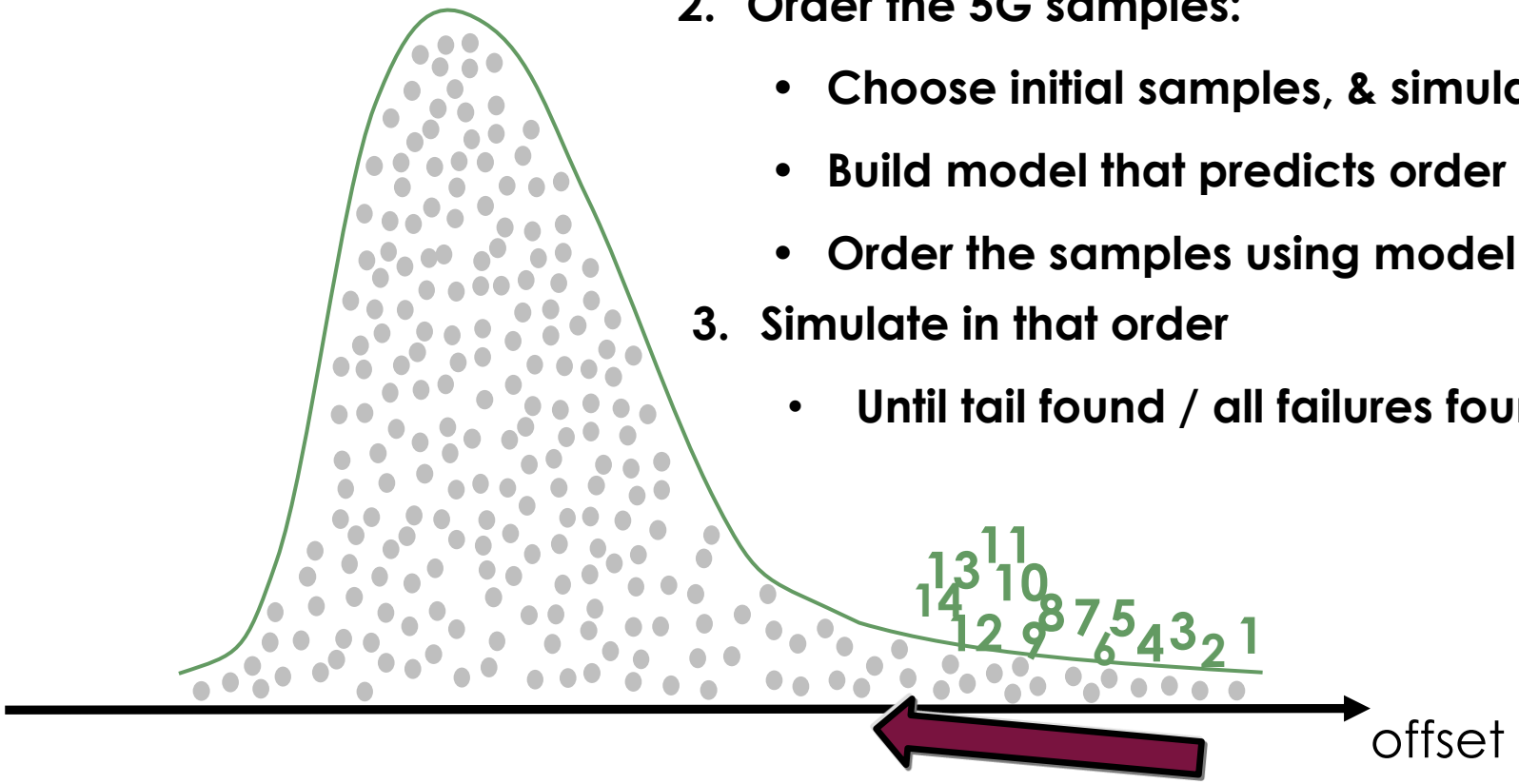
Make this *great*
e.g. High-Sigma Monte Carlo
(100x+ faster than plain MC)



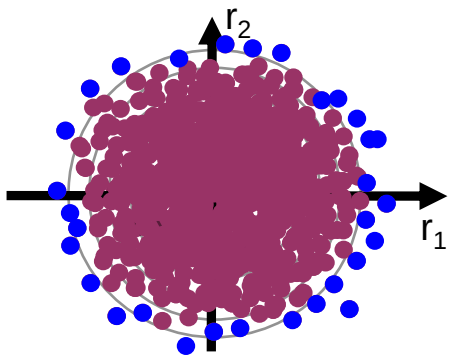


High-Sigma Monte Carlo Algorithm
Exploits scalability of Monte Carlo

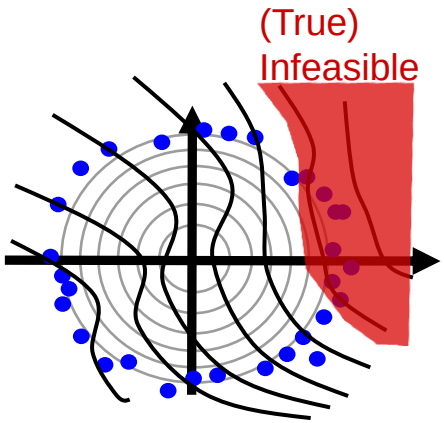
1. Generate (but don't simulate) 5G Monte Carlo samples
2. Order the 5G samples:
   - Choose initial samples, & simulate
   - Build model that predicts order
   - Order the samples using model
3. Simulate in that order
   - Until tail found / all failures found



High-Sigma Monte Carlo Algorithm
Details



- Generate (but don't simulate) 1B Monte Carlo samples.
- Choose initial samples = the 1000 generated samples that are farthest from nominal.
- Simulate initial samples.

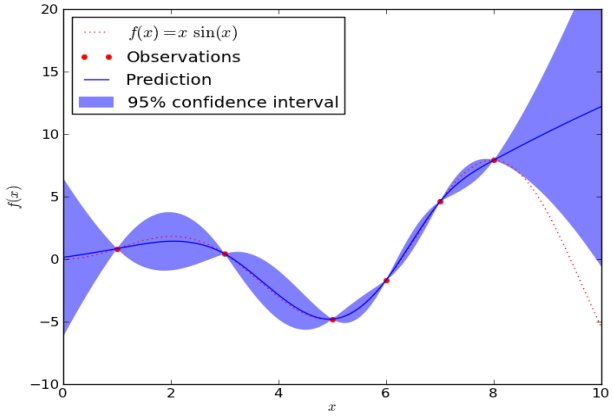


Contour lines are model-predicted output values.

- Build model that predicts order, mapping process points → simulated output values.

- Order the samples using model. Simulate the points in order, until all failures found or max # sims hit.

Cell Optimizer Algorithm
Truly global. Data-mines simulations to the max.



Cell Optimizer

Take initial samples

Build model mapping designs to performances

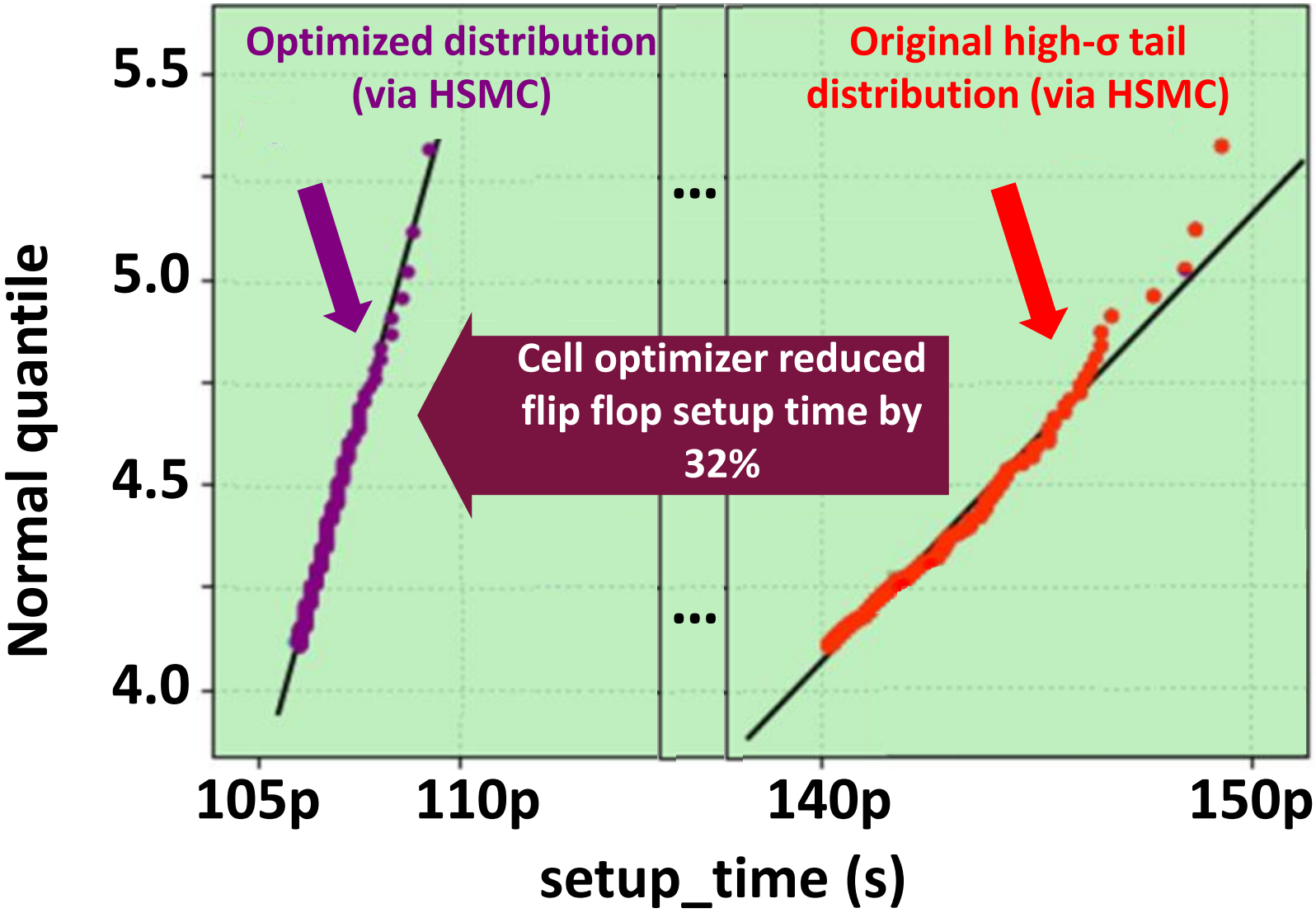
Choose new design point(s) via model

Simulate new points

Stop if converged

SPICE

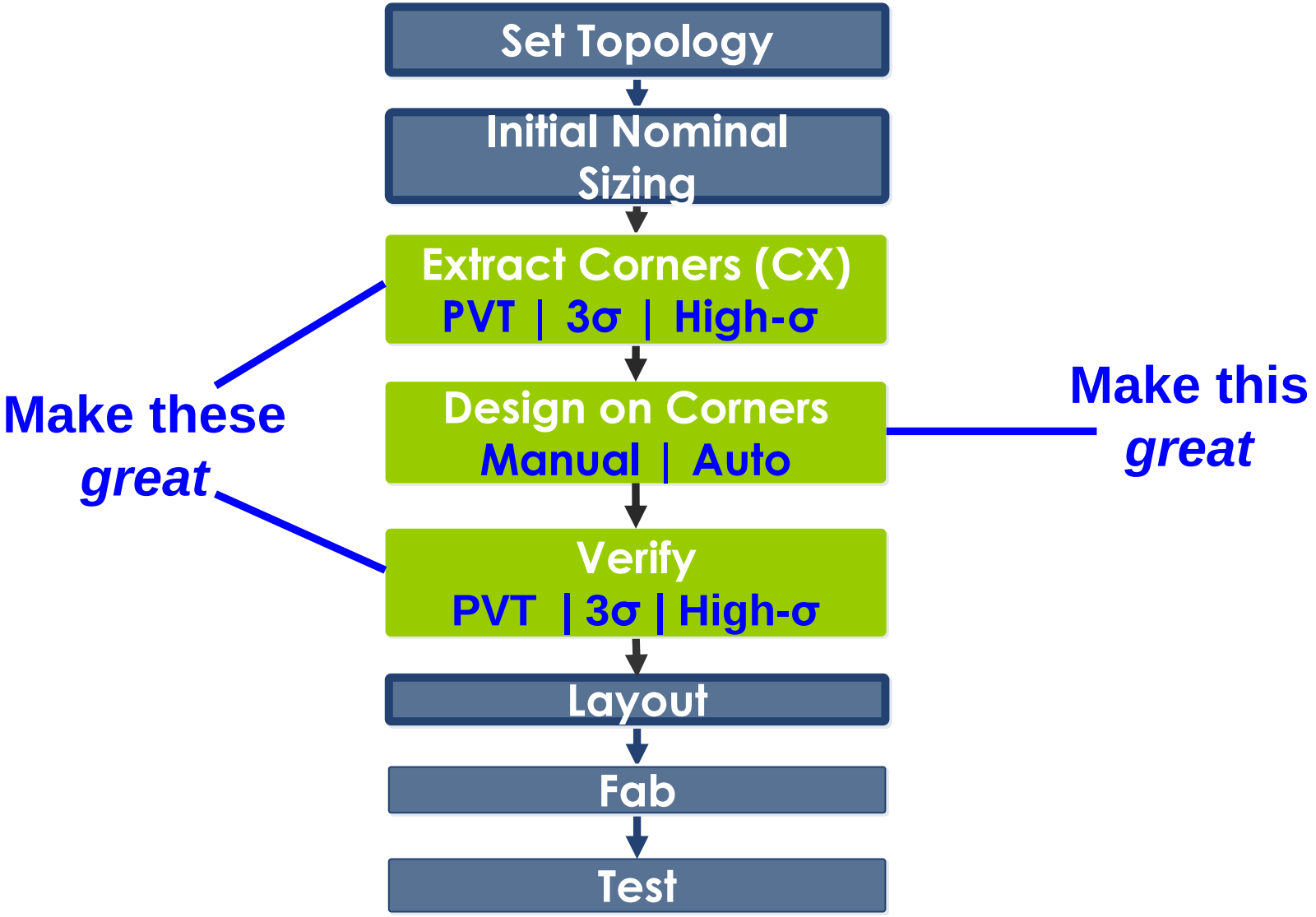
Example: minimize 5σ flip flop setup time
Using High-Sigma Monte Carlo (HSMC) and Cell Optimizer



Sigma-driven flow for memory and standard cells: Industrial user success stories

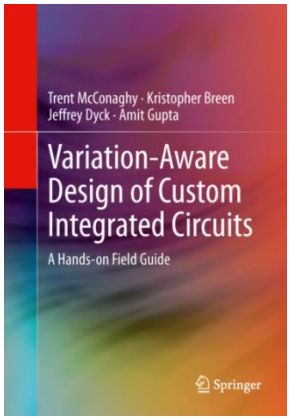
- **Efficiently optimizing a bitcell design for optimal read margin and write margin simultaneously**, using a different testbench for each measurement. This is for both nominal and high-sigma variation conditions. 20nm, 16nm.
- **Tuning, porting, retargeting, and migrating large standard cell libraries, to support a large designer base.** This task is extremely time consuming and expensive to do manually. Furthermore, these cells need to be optimized to work well under high-sigma conditions. 28nm, 20nm, 16nm.
- In a large library of standard cells, while all of the cells perform well under nominal conditions, **identifying which cells fail under high-sigma conditions and automatically fixing them via resizing**.

Sigma-Driven Corners: A *Unified* Flow For 3-σ, High-σ, and even PVT Variation



Conclusion

- Variation is a huge problem.
- Analog analysis includes memory and standard cells
- The sigma-driven corners flow is a unified way to handle variation. Fast, effective, familiar.
- For memory and standard cells, the most effective variant has:
  - Fast high-sigma analysis. E.g. High-Sigma Monte Carlo.
  - Fast global optimization. E.g. Cell Optimizer.



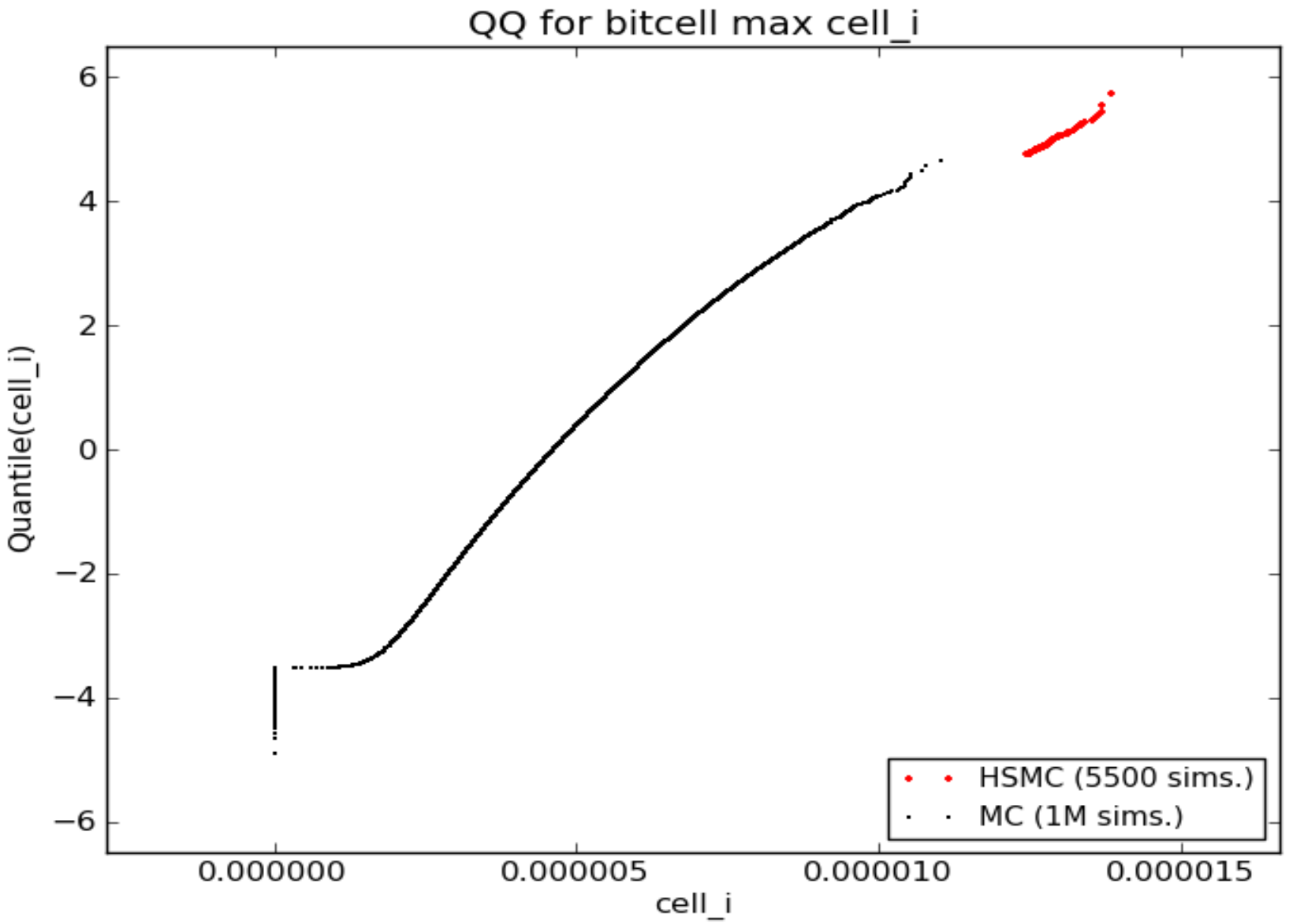
More detail:

T. McConaghy, K. Breen, J. Dyck, and A. Gupta. Variation-Aware Design of Custom Integrated Circuits: A Hands-on Field Guide. Springer, 2012.

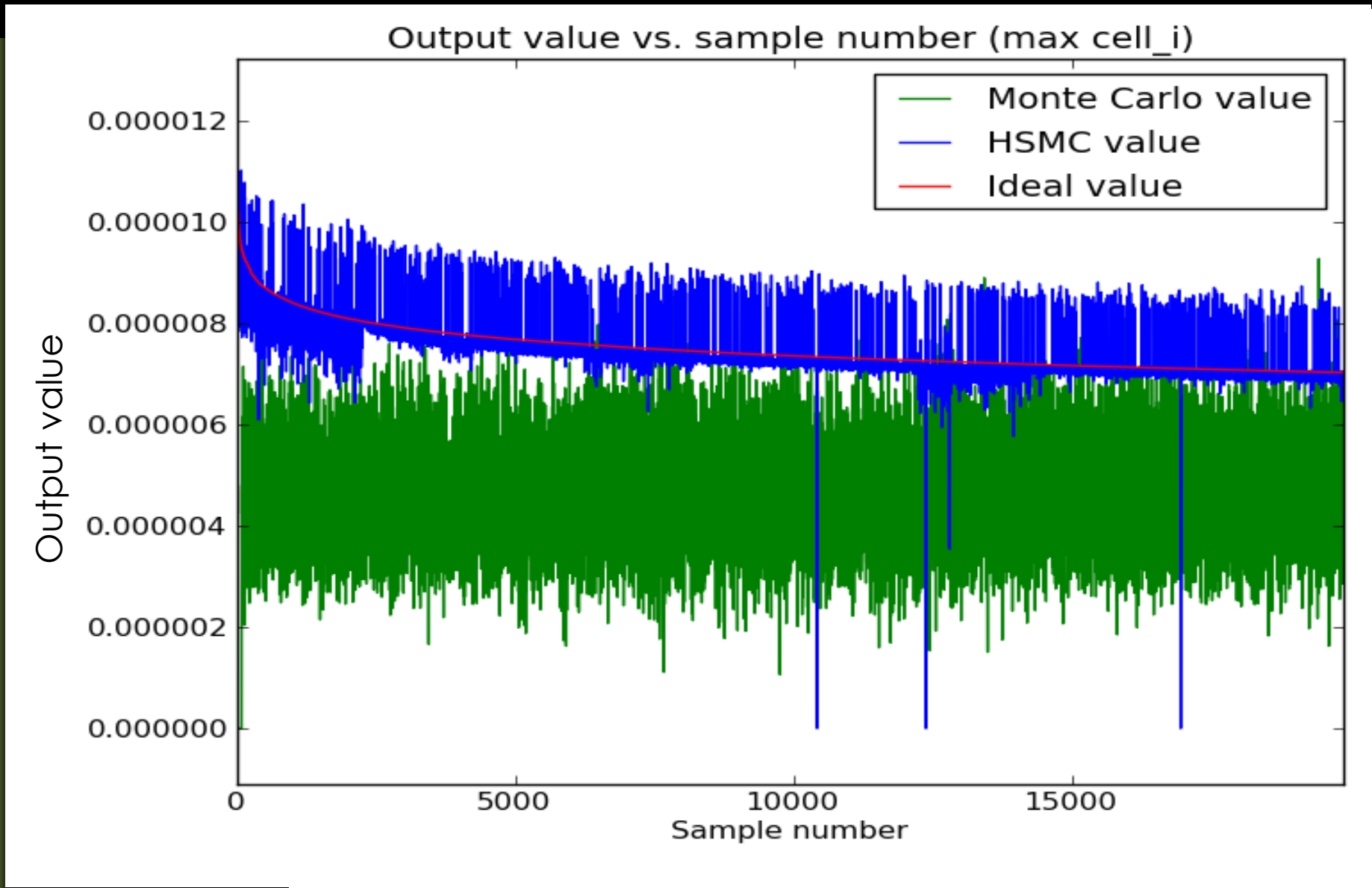
Extra slides

solido DESIGN AUTOMATION

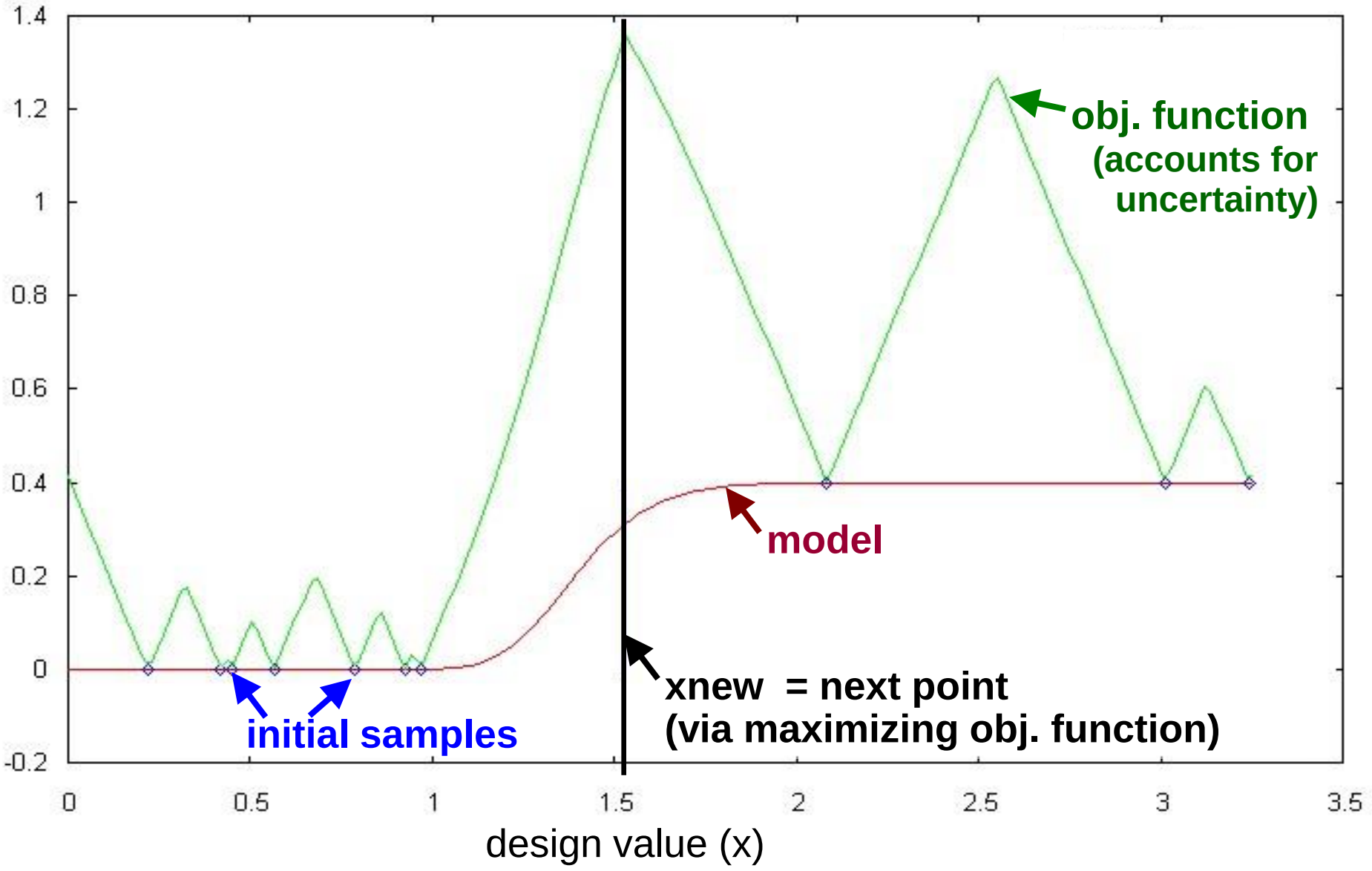
HSMC on Bitcell cell_i, w/ 100M MC samples
Shows: Fast, Accurate, Scalable



HSMC Convergence Curve on Bitcell cell_i
Shows: Verifiable ("How can I trust it?")



Cell Optimizer: Example Step 1
Initial sampling, simulate, build model, Choose xnew



Cell Optimizer: Example Step 2
Simulate new point, update model, choose xnew