From Finance to Flip Flops: Using the Mathematics of Money and Risk to Understand the Statistics of Nanoscale Circuits

Rob A. Rutenbar (and Amith Singhee)
Professor, Electrical & Computer Engineering
rutenbar@ece.cmu.edu (and asinghe@us.ibm.com)

About This Talk

- Statistics for nanoscale ckt
t  - The new challenge
- Monte Carlo analysis
  - How we do statistical analysis
- Mathematics of money+risk
  - Surprising source for very sophisticated Monte Carlo tools
New Challenge: **Statistical Variation**

- At nanoscale, nothing is *deterministic* anymore
- Everything is *statistical*

**Statistical Variability: Two Flavors**

- **Systematic variation**
  - Ex: Lithography
  - Optics, chemistry to print small mask shapes
  - Not really random
    - Physics is understood, expensive to compute

- **Random variation**
  - Ex: Dopant fluctuation
  - How many individual dopant atoms; where?
  - Really (really) random
    - Physics is fundamentally random for these effects
End Result for Us Design/CAD Folks

Welcome to design in the nanoscale regime

To Evaluate Circuit Impact: **Monte Carlo**

- Sample each statistical variable
- Parameterize one circuit, simulate it
- Repeat--n samples
Monte Carlo Math: Just A Big Integral

\[ \text{Est} = \int \cdots \int f_{\text{ckt}}(x) p(x) \, dx \]

\[ \approx (1/n) \sum \hat{F}(u) \]

Can transform to sample uniformly from s-dim unit cube

Evaluate Circuit Impact: Monte Carlo

- **PRO:** Accurate, flexible, general
- **CON:** Slow, slow, slow...

Uniform random s-dim sample
Why is Monte Carlo Painful?

- High-dim problems: $s$ is big (100-1000)
- Profoundly nonlinear: Nanoscale physics
- Accuracy matters: $\sim$1-5% error
- Speed matters: Many samples
- Samples expensive: Simulate each circuit

Question: Who Else Has This Problem?

Computational finance

- Valuing complex financial instruments, derivatives
- High-dimensional, nonlinear, statistical integrals
- Speed+accuracy matters here, e.g., $\sim$real-time decision-making
A Brief Aside: About the “F” Word…

- These recruiting signs common in my building at CMU…

- ...last year

~1 year ago…

- “Wow, analyzing yield is like pricing a bond? Cool!”

~1 month ago…

- “Wow, you’re using the same stuff that killed Wall Street?!”
A Brief Aside: About the “F” Word…

“The people who ran the financial firms chose to program their risk-management systems with overly optimistic assumptions and to feed them oversimplified data. …

… Wall Street executives had lots of incentives to make sure their risk systems didn’t see much risk.”

From Finance to Physics…

- Moral of story: If you start with honest physics as your input, you can get great results…

RDF

0% down! 0% APR!
Monte Carlo Revisited: Uniform Sampling

2-D example: unit cube is $[0,1]^2$

- Independent, uniform random samples $(x,y)$ in 2-D cube

- Classical Monte Carlo sampling
  - Uses uniform pseudo-random pts (i.e, `rand()`)  
  - Surprise: *Not* very uniform (clumps, holes, etc)  
  - Turns out this is inefficient – we can do *better*

Better == Low Discrepancy

**High-discrepancy samples** vs **Low-discrepancy samples**

Mathematically: the *discrepancy* is a measure of “uniformity”

$$D_n^* = \sup_J \frac{n_J}{n} - Vol(J)$$

- Fraction of points in $J$  
- Fraction of volume occupied by $J$

**How well does sampled $n_J/n$ approximate relative volume of box?**

For *low-discrepancy* sequences, answer is: *always very well.*
Doing Better: Quasi Monte Carlo (QMC)

- Classical Monte Carlo
  - Uniform pseudo-random pts
  - Problem: not very uniform

- Quasi Monte Carlo
  - “Low-discrepancy” seq’s
  - Deterministic samples

- Error for n samples
  - $O(1 / \sqrt{n})$

Computational Finance Example

- Eval 5-year discount price for a bond
  - From [Ninomiya, Tezuka, App Math Finance 1996]

Error (log scale)

- 1439 dimensions
- Ideal ~ $1/\sqrt{n}$
- Monte Carlo
- Ideal ~ $1/n$
- QMC
- 150x faster
Engineering Detail: Pattern Artifacts

- Problem: Low Discrep Seq’s show patterns in high dim’s
  - Need too many points for good uniformity

- Solution: Since earlier dimensions less affected…
  - Calculate statistical sensitivity of all vars
  - Put sensitive vars first
  - Ex: in \( f(x_1, x_2) \) if \( x_1 \) more important than \( x_2 \)

Does QMC Work for Circuits?

- Yes!
  - See: [Singhee, Rutenbar, ISQED 2007]
  - Example: Complete SRAM column @ 90nm
Very Promising Speedups

- Same 403-dimensional, 64b SRAM column

![Graph showing log(n) vs log(Std Dev) with Monte Carlo and QMC, indicating ~9x faster for 1% error.]

[Singhee, Rutenbar, ISQED 2007]

Isn’t This Just Latin Hypercube Sampling?

- No
- LHS sample set actually a randomized low-discrep seq
- Considered “advanced” in EDA, but inferior to QMC
- (Nobody prices bonds with LHS – it’s all QMC)

Plots: Error (est. variance across 10 runs) vs #samples n
Good Behavior on Other Circuits

- MC vs QMC variance, with #samples, \( n \)
  - 10 MC runs to compute MC variance
  - 1 QMC run + 9 scrambled QMC runs for variance
- General result: See speedups of 2X – 50X

- Master-Slave FF + Scan
  - 45nm, 31 parameters
- 0.6V Bandgap Ref
  - 90nm, 122 parameters
- 64b SRAM Column
  - 90nm, 403 parameters

Are We Done Yet? (Nope...)

- Lots of ideas to exploit in this space
- Money...
- Risk
Another Aside: About the “F” Word…

“In fact, most Wall Street computer models radically underestimated the risk of the complex mortgage securities … partly because the level of financial distress is ‘the equivalent of the 100-year flood’…”

Next Problem: “Rare Event” Statistics

- SRAM reliability is all about far tails of stats
  - Why? High replication (~10^8 bits) of core circuits
  - 3σ doesn’t cut it for 100M cells; need 6σ, 7σ, 8σ…

Problem: Intractable Monte Carlo runs

- 1M Monte Carlo sims predicts (unreliably) to ~4.5σ
What Do We Need To Solve This…?

- **Ultra fast sampling** of rare events
  - Put Monte Carlo **samples** out in far tails -- directly

- **Accurate analytical pdf models** of rare tails
  - Using these samples, model lets us predict **farther**

Efficiently Sampling **Just** the Tail

- **Note:** *Generating* MC samples is cheap, *Simulating* these samples is costly

- **Idea:**
  1. *Generate* regular MC samples…
  2. …but *block* points that are “very probably” **not** in tail
  3. *Simulate* the rest – i.e., the points we do not block

Can build this **classifier** filter very efficiently
We Call the Idea: **Statistical Blockade**

- Simulate starting set (few points, fast)
- Sample points

**Circuit pdf**

<table>
<thead>
<tr>
<th>Tail</th>
<th>t (99%)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Build <strong>classifier</strong> (fast)</th>
</tr>
</thead>
</table>
| (uses ideas from data-mining; we use a **Support Vector Machine**)

<table>
<thead>
<tr>
<th>Set the classification threshold with some safety margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>t_c (97%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Generate MC samples (fast)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classify sample points (fast)</td>
</tr>
<tr>
<td>Block nontail points (fast)</td>
</tr>
<tr>
<td>Simulate the rest (slow)</td>
</tr>
</tbody>
</table>

**Set the classification threshold with some safety margin**

<table>
<thead>
<tr>
<th>t_c t</th>
</tr>
</thead>
</table>

---

**Modeling Statistics of Rare Events...?**

- **Extreme Value Theory (EVT)**
  - Behavior of extreme (rare) values of distributions
  - (If hurricanes are i.i.d random variables, we’d like to know the statistics of the largest waves...)
EVT: Modeling the PDF in the Tail

- Recall Central Limit Theorem: $\Sigma$ (i.i.d. samples) $\rightarrow$ Gaussian
- Question: Is there a similar result for these tails of “extreme” results ...?
- Answer: YES – Extreme Value Theory (EVT)

On each of $N$ wafers, identify cells slower than threshold $t$. What is their distrib? EVT tells us!

- EVT gives simple analytical form for conditional tail distrib

$$G_{\alpha}(x) = \begin{cases} 1 - \left(1 - \frac{kx}{\alpha}\right)^{-\frac{1}{k}}, & k \neq 0 \\ 1 - e^{-\frac{x}{\alpha}}, & k = 0 \end{cases}$$

Result: Complete 64b SRAM Column

- 90nm 64b SRAM column with write driver and column mux
  - ~ 400 devices; model Write-time CDF
  - Speedup: ~16X

- Std Monte Carlo: 100,000 sims
- Statistical Blockade: 6,314 sims
  - 1000 sims to build classifier
  - 100,000 points $\rightarrow$ 5314 sims
Result: Validating Model Out to 8σ

- Recently validated novel analytical DRV model
  - Model of Data Retention Voltage, [Calhoun et al. UVa, ESSCIRC'07]
  - Validated to 6σ, via billion element Monte Carlo run…
  - …but only did 41,721 SPICE sims – recursive extension of Blockade
  - Speedup ~23,000X

At nanoscale, nothing is deterministic…

Brute-force Monte Carlo hurts (a lot)

We can do much better with smart methods
  - (Many of which involve $$$ + risk…)
  - CMU results: 10x – 10,000x speedups
Thank You!

Acknowledgements

- My former student, Dr. Amith Singhee, whose PhD is the basis of all the results shown in this talk

- Prof. Benton Calhoun and Jiajing Wang of U Virginia, for sharing their statistical DRV model

- Funding from Semiconductor Research Corporation

- Funding from the Focus Center for Circuit & System Solutions (C2S2), one of five such focus centers managed by the Focus Center Research Program, an SRC program.