Stochastic Analog Circuit Behavior Modeling by Point Estimation Method

Fang Gong\textsuperscript{1}, Hao Yu\textsuperscript{2}, Lei He\textsuperscript{1}

\textsuperscript{1}Univ. of California, Los Angeles
\textsuperscript{2}Nanyang Technological University, Singapore
Outline

- Backgrounds
- Existing Methods and Limitations
- Proposed Algorithms
- Experimental Results
- Conclusions
IC Technology Scaling

- Feature size keeps scaling down to 45nm and below
  - 90nm
  - 65nm
  - 45nm

- Large process variation lead to *circuit failures* and *yield problem*.

* Data Source: Dr. Ralf Sommer, DATE 2006, COM BTS DAT DF AMF;
Statistical Problems in IC Technology

- Statistical methods were proposed to address variation problems.
- Focus on **performance probability distribution extraction** in this work.

How to model the stochastic circuit behavior (performance)?
Leakage Power Distribution

- An example ISCAS-85 benchmark circuit:
  - all threshold voltages (Vth) of MOSFETs have variations that follow normal distribution.

- The leakage power distribution follow lognormal distribution.

- It is desired to extract the arbitrary (usually non-normal) distribution of performance exactly.

Problem Formulation

- **Given**: random variables in **parameter space**
  - a set of (normal) random variables \( \{\epsilon_1, \epsilon_2, \epsilon_3, \ldots\} \) to model process variation sources.

- **Goal**: extract the arbitrary probability distribution of performance \( f(\epsilon_1, \epsilon_2, \epsilon_3, \ldots) \) in **performance space**.
Outline

- Backgrounds
- **Existing Methods and Limitations**
- Proposed Algorithms
- Experimental Results
- Conclusions
Monte Carlo simulation

- Monte Carlo simulation is the most straight-forward method.

- However, it is highly time-consuming!
Response Surface Model (RSM)

- Approximate circuit performance (e.g. delay) as an analytical function of all process variations (e.g. \( \Delta V_{TH} \), etc.)
  - Synthesize analytical function of performance as random variations.
  - Results in a multi-dimensional model fitting problem.

- Response surface model can be used to
  - Estimate performance variability
  - Identify critical variation sources
  - Extract worst-case performance corner
  - Etc.

\[ f(\varepsilon) = p_0 + \alpha_1 \varepsilon_1 + \cdots + \alpha_N \varepsilon_N \]
Flow Chart of APEX*

Synthesize analytical function of performance using RSM

Calculate moments

Calculate the probability distribution function (PDF) of performance based on RSM

\[ f(\varepsilon) = p_0 + \alpha_1 \varepsilon_1 + \cdots + \alpha_N \varepsilon_N \]

\[ m_f^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} f^k \cdot pdf(f) \, df \]

\[ m_t^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} t^k \cdot h(t) \, dt \]

\[ h(t) = \begin{cases} 
\sum_{r=1}^{M} a_r \cdot e^{b_r^{k+1} \cdot t} & (t \geq 0) \\
0 & (t < 0) 
\end{cases} \]

*h(t) can be used to estimate \( pdf(f) \)

Limitation of APEX

- RSM based method is time-consuming to get the analytical function of performance.
  - It has exponential complexity with the number of variable parameters $n$ and order of polynomial function $q$.
    
    \[ f(x_1, x_2, \ldots, x_n) = (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n)^q \]

  - e.g., for 10,000 variables, APEX requires 10,000 simulations for linear function, and 100 millions simulations for quadratic function.

- RSM based high-order moments calculation has high complexity
  - the number of terms in $f^k$ increases exponentially with the order of moments.
    
    \[ E(f^p) = \int_{-\infty}^{+\infty} (f^p \cdot p df(f))df \]

    \[ f^k(x_1, x_2, \ldots, x_n) = (\alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n)^{k \times q} \]
## Contribution of Our Work

### Step 1: Calculate High Order Moments of Performance

<table>
<thead>
<tr>
<th>APEX</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Find analytical function of performance using RSM $f(\varepsilon) = p_0 + \alpha_1 \varepsilon_1 + \cdots + \alpha_N \varepsilon_N$</td>
<td>A few samplings at selected points.</td>
</tr>
<tr>
<td>Calculate high order moments $m_f^k = \int_{-\infty}^{+\infty} (f^k \cdot pdf(f))df$</td>
<td>Calculate moments by Point Estimation Method</td>
</tr>
</tbody>
</table>

### Step 2: Extract the PDF of performance

$$m_f^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} (f^k \cdot pdf(f))df = m_f^k = \frac{(-1)^k}{k!} \cdot \int_{-\infty}^{+\infty} (t^k \cdot h(t))dt = -\sum_{r=1}^{M} \frac{a_r b_r^{k+1}}{b_r}$$

$h(t) = \sum_{r=1}^{M} a_r \cdot e^{b_r t} \approx pdf(f)$

- **Our contribution:**
  - We do **NOT** need to use analytical formula in RSM;
  - Calculate high-order moments efficiently using **Point Estimation Method**;
Outline

- Backgrounds
- Existing Methods and Limitations
- Proposed Algorithms
- Experimental Results
- Conclusions
Moments via Point Estimation

- Point Estimation: approximate high order moments with a weighted sum of sampling values of $f(x)$.
  - $x_{j} \ (j = 1, \cdots, p)$ are estimating points of random variable.
  - $P_{j}$ are corresponding weights.
  - $k$-th moment of $f(x)$ can be estimated with

$$m_{k} = \int_{-\infty}^{+\infty} f^{k} \cdot pdf(f) \, df \approx \sum_{j=1}^{p} P_{j} \cdot f(x_{j})^{k}.$$

- Existing work in mechanical area* only provide empirical analytical formulae for $x_{j}$ and $P_{j}$ for first four moments.


Question – how can we accurately and efficiently calculate the higher order moments of $f(x)$?
Calculate moments of performance

- Theorem in Probability: assume $x$ and $f(x)$ are both continuous random variables, then:

$$E(f^k(x)) = \int f^k(x) \cdot pdf(f) \, df = \int f^k(x) \cdot pdf(x) \, dx$$

- Flow Chart to calculate high order moments of performance:

1. **Step 1**: Calculate moments of parameters
   $$m_x^k = \int_{-\infty}^{+\infty} (x^k \cdot pdf(x)) \, dx \approx \sum_{j=1}^{m} P_j \cdot (x_j)^k$$

2. **Step 2**: Calculate the estimating points $x_j$ and weights $P_j$

3. **Step 3**: Run simulation at estimating points $x_j$ and get performance samplings $f(x_j)$

4. **Step 4**: Calculate moments of performance
   $$m_f^k = \int_{-\infty}^{+\infty} (f^k \cdot pdf(x)) \, dx \approx \sum_{j=1}^{m} P_j \cdot (f(x_j))^k$$

5. **Step 5**: Extract performance distribution $pdf(f)$

**Step 2 is the most important step in this process.**
Estimating Points $x_j$ and Weights $P_j$

- With moment matching method, $x_j$ and $P_j$ can be calculated by
  \[
  \sum_{j=1}^{m} P_j = 1 = m_x
  \]
  \[
  \sum_{j=1}^{m} P_j \cdot x_j = E(x) = m_1
  \]
  \[
  \sum_{j=1}^{m} P_j \cdot x_j^2 = E(x^2) = m_2
  \]
  \[\vdots\]
  \[
  \sum_{j=1}^{m} P_j \cdot x_j^{2m-1} = E(x^{2m-1}) = m_{2m-1}
  \]

  \[m_x^k (k = 0, \ldots, 2m - 1)\] can be calculated exactly with $pdf(x)$.

- Assume residues $a_j = P_j$ and poles $b_j = 1 / x_j$

  \[
  \begin{bmatrix}
  a_1 + a_2 + \cdots + a_m \\
  \frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_m}{b_m} \\
  \vdots \\
  \frac{a_1}{b_1^{2m-1}} + \frac{a_2}{b_2^{2m-1}} + \cdots + \frac{a_m}{b_m^{2m-1}}
  \end{bmatrix}
  =
  \begin{bmatrix}
  m_x^0 \\
  m_x^1 \\
  m_x^2 \\
  \vdots \\
  m_x^{2m-1}
  \end{bmatrix}
  \]

- System matrix is well-structured (Vandermonde matrix);
- Nonlinear system can be solved with deterministic method.
Extension to Multiple Parameters

- Model moments with multiple parameters as a linear combination of moments with single parameter.

\[ m_f^{k}(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} g_i m_f^{k}(x_i) \]

\[ g_i = c \cdot \frac{\partial (f(x_i))}{\partial x_i} \]

\[ c = 1 / \sqrt{\sum_{i=1}^{n} \frac{\partial (f(x_i))}{\partial x_i}} \]

- \( f(x_1, x_2, \ldots, x_n) \) is the function with multiple parameters.
- \( f(x_i) \) is the function where \( x_i \) is the single parameter.
- \( g_i \) is the weight for moments of \( f(x_i) \)
- \( c \) is a scaling constant.
Error Estimation

- We use approximation with $q+1$ moments as the exact value, when investigating PDF extracted with $q$ moments.

- When moments decrease progressively $|m_f^p| \geq |m_f^{q+1}|$ ($p \leq q + 1$)

\[
m_f^k = \int_{-\infty}^{+\infty} (f^k \cdot pdf(f))df
\]

\[
Error \leq \left| \frac{(-j\omega)^{q+1}}{(q+1)!} \cdot \left( \sum_{\nu=0}^{q+1} \frac{(-j\omega)^p}{p!} \right)^{-1} \right|
\]

- Other cases can be handled after shift ($f<0$), reciprocal ($f>1$) or scaling operations of performance merits.
Outline

- Backgrounds
- Existing Methods and Limitations
- Proposed Algorithms
- Experimental Results
- Conclusions
(1) Validate Accuracy: Settings

- To validate accuracy, we compare following methods:
  - Monte Carlo simulation.
    - run tons of SPICE simulations to get performance distribution.
  - PEM: point estimation based method (proposed in this work)
    - calculate high order moments with point estimation.
  - MMC+APEX:
    - obtain the high order moments from Monte Carlo simulation.
    - perform APEX analysis flow with these high-order moments.

\[
\hat{m}_f^k = \frac{(-1)^k}{k!} \int_{-\infty}^{+\infty} f^k \cdot pdf(f) \, df
\]

\[
\hat{m}_f^k = \hat{m}_z^k = - \sum_{r=1}^{M} \frac{a_r}{b_r^{k+1}}
\]

\[
h(t) = \begin{cases} 
  \sum_{r=1}^{M} a_r \cdot e^{b_r t} & (t \geq 0) \\
  0 & (t < 0)
\end{cases}
\]
6-T SRAM Cell

- Study the discharge behavior in $BL_B$ node during reading operation.
- Consider threshold voltage of all MOSFETs as independent Gaussian variables with 30% perturbation from nominal values.
- Performance merit is the voltage difference between $BL$ and $BL_B$ nodes.
Accuracy Comparison

- Variations in threshold voltage lead to deviations on discharge behavior.
  - Investigate distribution of node voltage at certain time-step.
- Monte Carlo simulation is used as baseline.
- Both APEX and PEM can provide high accuracy when compared with MC simulation.
(2) Validate Efficiency: PEM vs. MC

- For 6-T SRAM Cell, Monte Carlo methods requires $3000$ times simulations to achieve an accuracy of 0.1%.

- Point Estimation based Method (PEM) needs only $25$ times simulations, and achieve up to $119X$ speedup over MC with the similar accuracy.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (second)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monte Carlo ($3 \times 10^3$)</td>
<td>7644</td>
<td>1x</td>
</tr>
<tr>
<td>PEM (5 point)</td>
<td>64.12</td>
<td>119.9x</td>
</tr>
</tbody>
</table>
Compare Efficiency: PEM vs. APEX

- To compare with APEX:
  - One Operational Amplifier under a commercial 65nm CMOS process.
  - Each transistor needs 10 independent variables to model the random variation*.

<table>
<thead>
<tr>
<th>Circuit Name</th>
<th>Transistor #</th>
<th>Mismatch Variable #</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM Cell</td>
<td>~ 6</td>
<td>~ 60</td>
</tr>
<tr>
<td>Operational Amplifier</td>
<td>~ 50</td>
<td>~ 500</td>
</tr>
<tr>
<td>ADC</td>
<td>~ 2K</td>
<td>~ 20K</td>
</tr>
<tr>
<td>SRAM Critical Path</td>
<td>~ 20K</td>
<td>~ 200K</td>
</tr>
</tbody>
</table>

- We compare the efficiency between PEM and APEX by the required number of simulations.

- Linear vs. Exponential Complexity:
  - PEM: a linear function of number of sampling point and random variables.
  - APEX: an exponential function of polynomial order and number of variables.

Operational Amplifier

- A two-stage operational amplifier
  - Complexity in APEX increases exponentially with polynomial orders and number of variables.
  - PEM has linear complexity with the number of variables.

Operational Amplifier with 500 variables

- Quadratic polynomial case
  - The Y-axis in both figures has log scale!

The Y-axis in both figures has log scale!
Conclusion

- Studied **stochastic analog circuit behavior modeling under process variations**

- Leverage the **Point Estimation Method (PEM)** to estimate the high order moments of circuit behavior **systematically and efficiently**.

- Compared with exponential complexity in APEX, proposed method can achieve **linear complexity** of random variables.
Thank you!

ACM International Symposium on Physical Design 2011

Fang Gong, Hao Yu and Lei He