Elements of Statistical SPICE Models

N. Lu, J. Watts, S. Springer

IBM

Semiconductor Research & Dev. Center

Burlington, Vermont, USA

Outline

- Requirements for Statistical SPICE Models
- Elements in Statistical SPICE Models
  - 1. Distinguishing Monte Carlo vs. corner simulations
  - 2. Modeling of asymmetric statistical distributions
  - 3. Treatment of many process statistical distributions in skewed/corner simulations
  - 4. Modeling the combined effects of chip-mean (systematic) and across-chip (random) variations
- Summary

Requirements for Statistical SPICE Models

- **Support both Monte Carlo (MC) and skewed/corner simulations**
  - **Best also support Across-Chip Variation (ACV) MC simulations at a chip-mean corner**
- **Able to handle severe asymmetric distributions**
  - **Best also to handle an arbitrary distribution which can be symmetric, moderately asymmetric, or severely asymmetric**
- **Provide both easy skewing (few skewing parameters) and full skewing (many skewing parameters; for optimal corner)**
- **Correctly combine global and local variations**

1. Distinguish MC vs. Corner Simulations

- **A typical method:**

```
.param mc_sw = 0    $ 0 or 1
+ rs_f = rs_n + rs_3sig * (mc_sw * agauss(0,1,3)
                             + (1 – mc_sw) * cor_res/3)
```

  - **Problem: Switch setting and actual netlist's MC or corner simulations are two different things. No guarantee that the two will be the same**

- **Another typical method:**

```
+.param cor_res = 0.0
+ rs_f = rs_n + rs_3sig * agauss(cor_res/3, 1, 3)
```

and ask users not to run MC with cor_res $\neq$ 0

  - **Problem: Does not prevent model's misuse**

Element 1. Our Solution

- **Automatic detection of MC vs. skewed/corner simulations**

```
.param mc1 = agauss(0, 1, 3)
+      mc2 = agauss(0, 1, 3)
+      mc3 = agauss(0, 1, 3)
+ is_mc = ‘mc1 != 0 || mc2 !=0 || mc3 !=0’
...
+ p1_f = ‘is_mc ? p1_mc : p1_skewing’
+ p2_f = ‘is_mc ? p2_mc : p2_skewing’
```

- **There are other automatic detection methods. For example:**

```
+ is_mc = ‘mc1 != mc2 || mc2 != mc3’
```

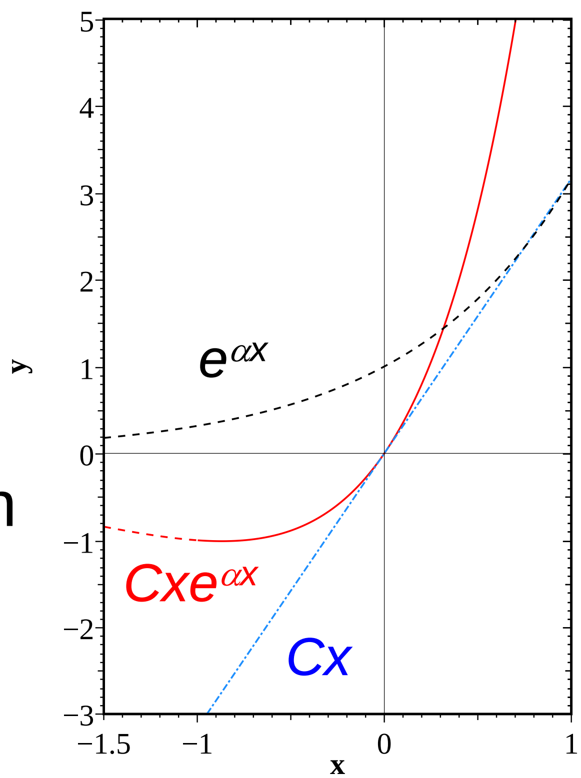
2. Modeling asymmetric distributions

- **Solution method: Use a mapping relation to transfer a symmetric distribution to an asymmetric one**
- **Requirements:**
  - **Mapping relation be monotonic between 3σ lower and 3σ upper bounds**
  - **Be able to model a severe asymmetric distribution**
- **Best solution:**
  - **Can handle an arbitrary distribution (symmetric, moderately asymmetric, or severely asymmetric) at run time**
    - **Usage: Statistical SPICE models for supporting parasitic R and C extraction tools (PEX tools)**
    - **Distributions of parasitic R and C are specified in a SPICE netlist by a PEX tool after the models are built**

Element 2. Existing solutions

- $y = x^2$, $y = x^3$, $y = e^x$, etc.:
  - There is an upper limit on (max – nom)/(nom – min)
  - Can not handle both symmetric and asymmetric distribution at run time

- $y = Cxe^{\alpha x}$, etc.:
  - Can handle both symmetric and asymmetric distributions
  - But is not always monotonic between min and max of a distribution

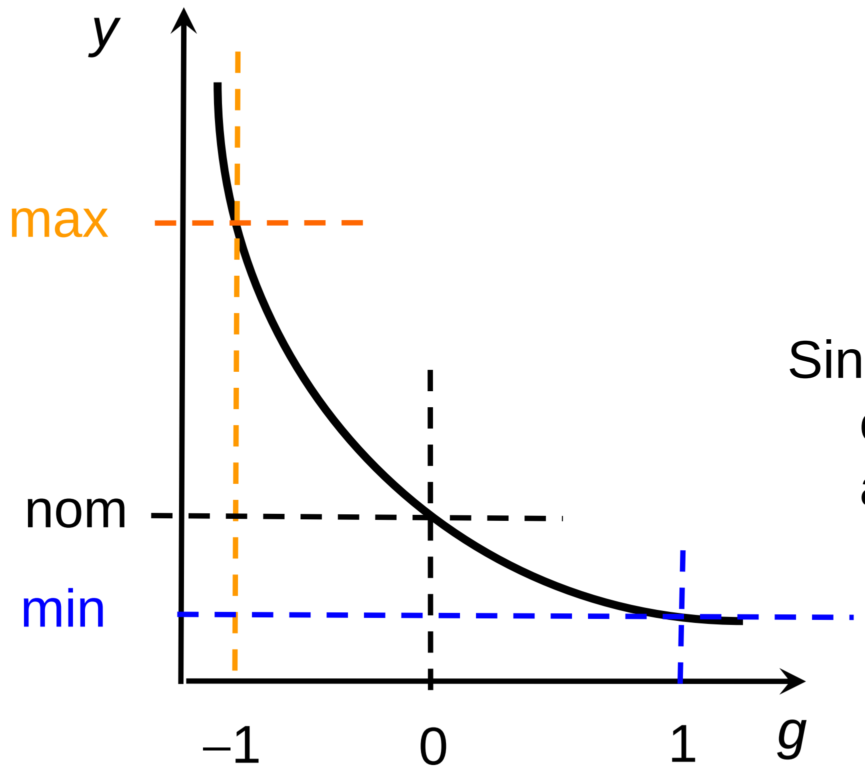


Element 2. Our Solution (1)

A simple mapping relation:

$$y(g; b, n, w) = n + C\left(\frac{1}{1+\alpha g} - 1\right)$$

Always monotonic



n – nominal value

b – 3σ best-case value (could be either max or min)

w – 3σ worst-case value (could be either min or max)

g – a symmetric distribution

C, α – Two coefficients

Singularity at (1+ αg) = 0 provides the capability of modeling a very asymmetric distribution

Element 2: Our Solution (2)

Its solution:

$$asym(g;b,n,w) = n + \frac{2(w-n)(b-n)g}{w-b+(w+b-2n)g}$$

is simple and powerful

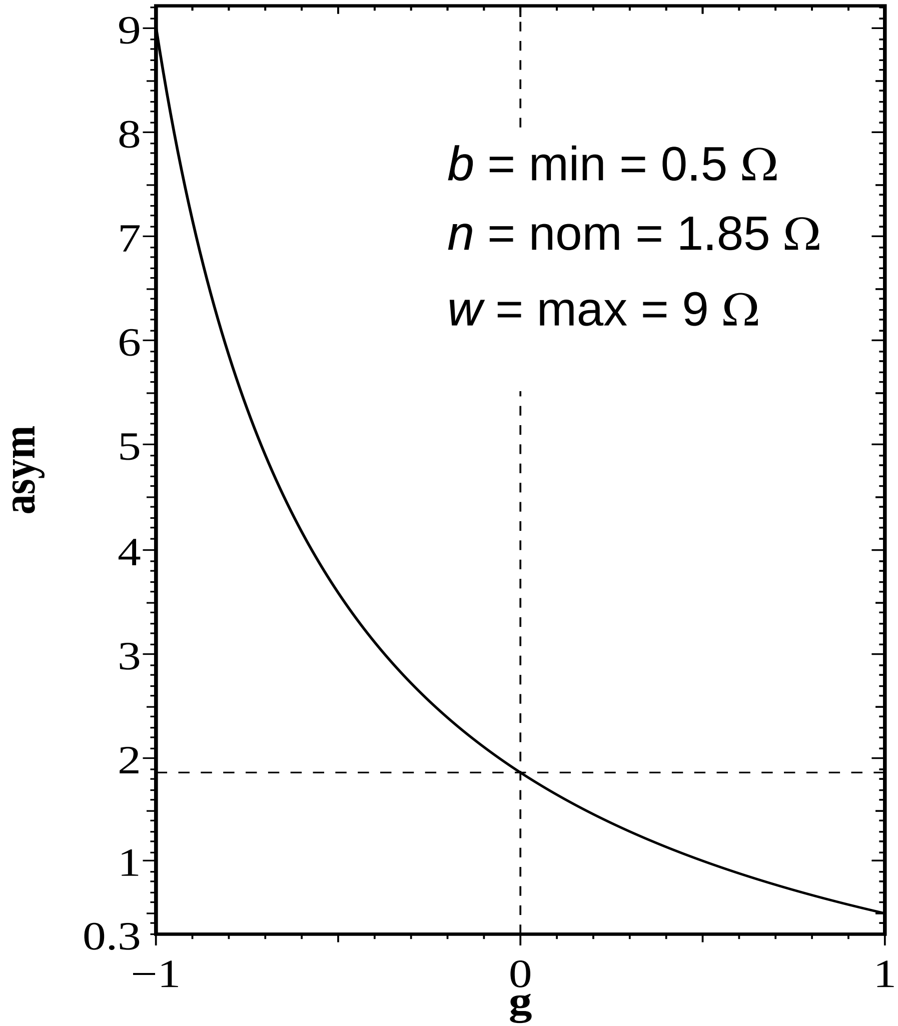
When a distribution is symmetric,

the above solution simplifies to:

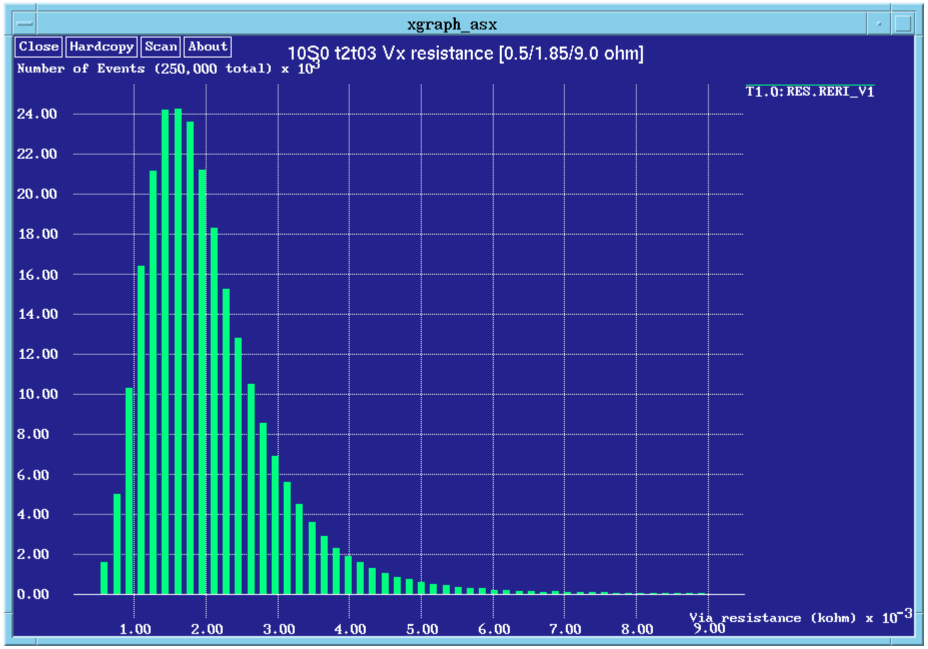
$$sym(g;b,n,w) = n + (b-n)g \;, \quad w+b = 2n$$

Element 2: Via resistance example

Mapping relation



Histogram of MC simulation



Element 2: Other Mapping Relations

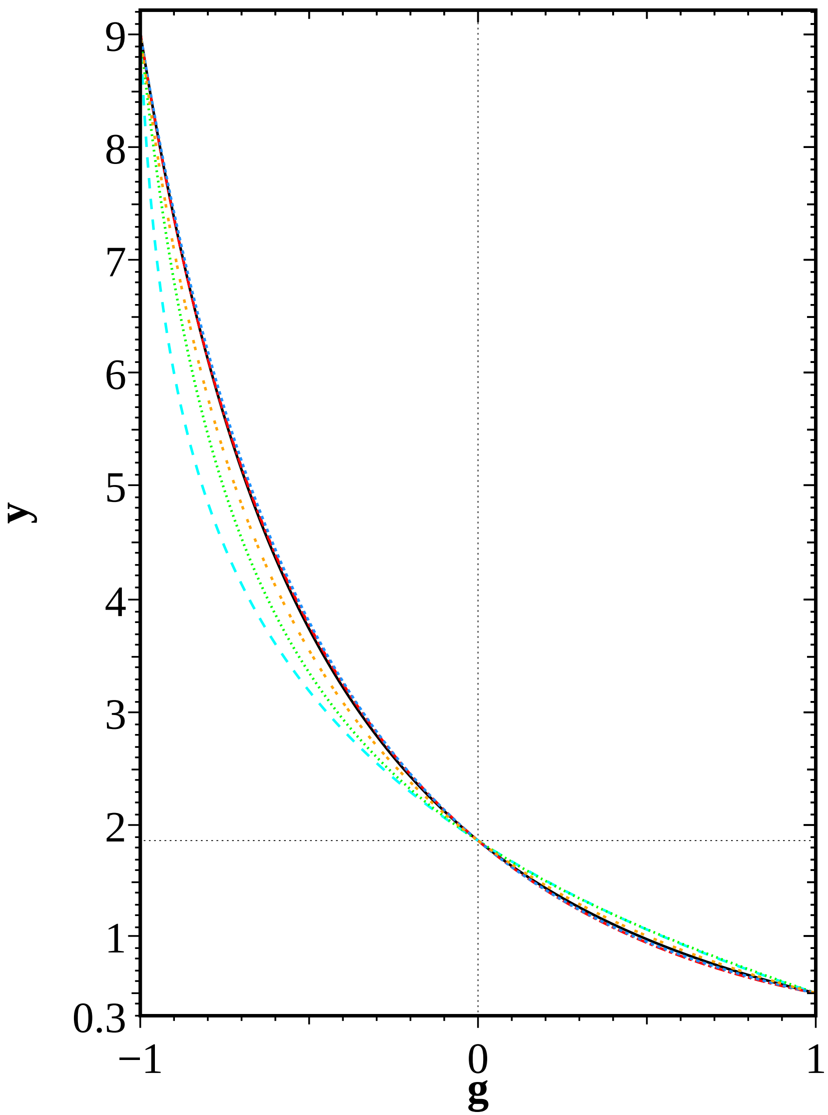
There are other forms of mapping relations to model asymmetric distributions

For the same via resistance example:

b = min = 0.5 Ω

n = nom = 1.85 Ω

w = max = 9 Ω

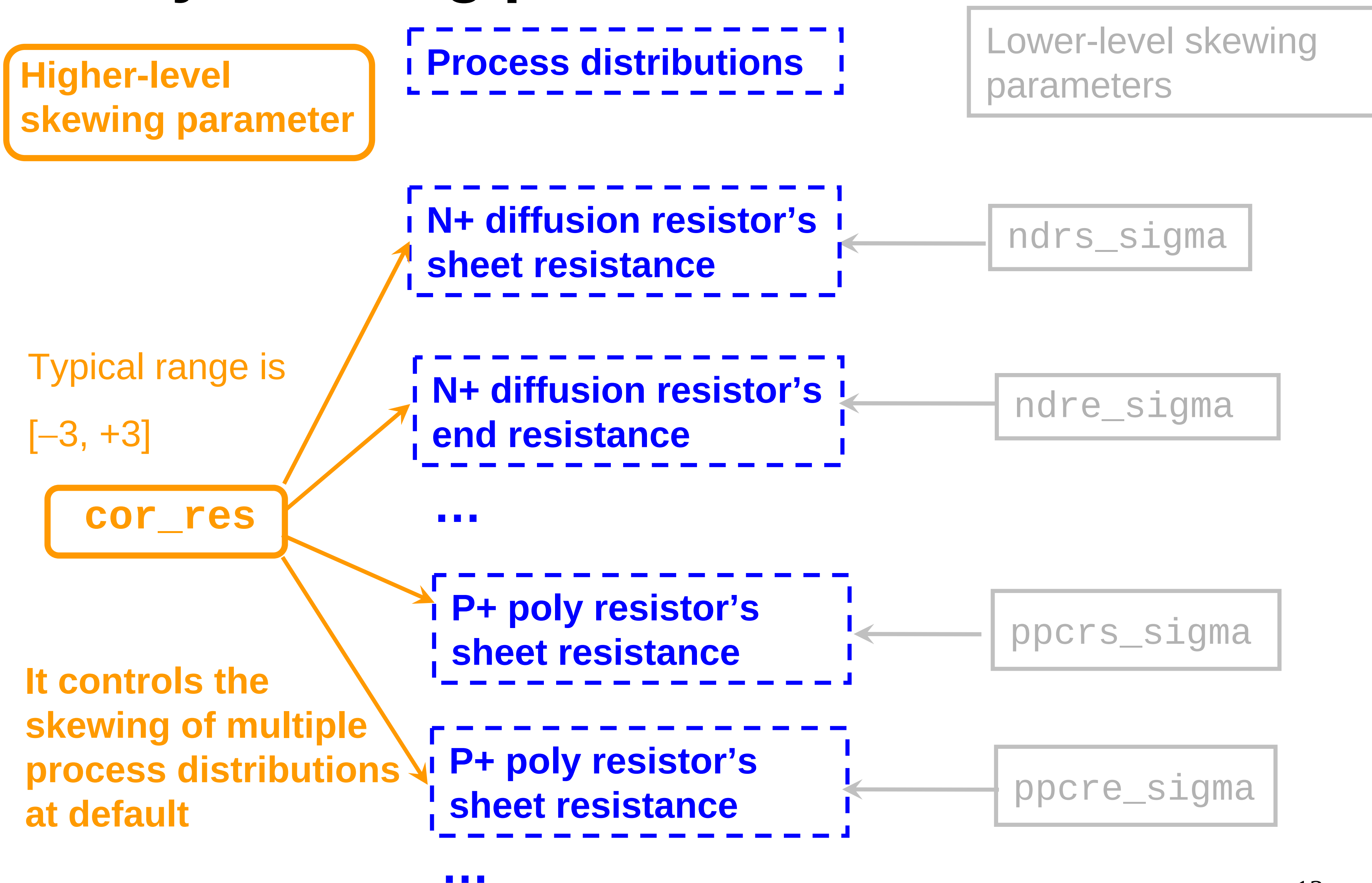


Explicit expressions are given in the paper

3. Treatment of Many Statistical Distributions in Corner Simulations

- **Two opposing requirements:**
  - **A. For ease of use, we prefer to have only a few skewing parameters to control many process distributions. e.g.,**
    - **`cor_res` for poly & diffusion resistors**
    - **`cor_cap` for junction & MIM capacitors**
    - **`cor_wirerc` for interconnect parasitic R and C**
  - **B. To find all sensitivities of a performance target to all process distributions, to reach an optimal corner, or to find a common corner of 2 or more performance targets, we also prefer to have an independent skewing parameter for each independent process distribution**

3. Our Solution: A hierarchical structure for many skewing parameters



3. Our Solution: A hierarchical structure (2)

Higher-level skewing parameter

Process distributions

Lower-level skewing parameters

N+ diffusion resistor's sheet resistance ← `ndrs_sigma=2.4`

N+ diffusion resistor's end resistance ← `ndre_sigma=1.8`

...

P+ poly resistor's sheet resistance ← `ppcrs_sigma=1.15`

P+ poly resistor's sheet resistance ← `ppcre_sigma=2.77`

...

`cor_res` → (all of the above process distributions)

If specified, a lower-level skewing parameter takes precedence over its higher-level skewing parameter

4. Modeling the Combined Effect of Chip-mean and Across-chip Variations

- **A. Combining chip-mean (CM) variation and across-chip variation (ACV) for a single model parameter**
  - **When ACV is treated as completely uncorrelated (e.g., Vth0)**

$$p_i = n + \sigma_{cm} G + \sigma_{acv} g_i \,, \quad i = 1, 2, 3, \cdots,$$

  - **When ACV is treated as completely correlated (e.g., channel length and width)**

$$p_i = n + \sigma_{cm} G + \sigma_{acv} g_0 \,, \quad i = 1, 2, 3, \cdots.$$

G, g_0, g_1, g_2, g_3, ... – stochastic/random variables

Total variation of *p*: $\sigma_p = \sqrt{\sigma_{cm}^2 + \sigma_{acv}^2}$

4. (Systematic + Random) Variations

- **B. When an ACV is characterized as a percentage of the mean (nominal) value:**

$$\sigma_{acv} = n\sigma_{mm}$$

Typical treatment:

$$p_i = (n + \sigma_{cm} G)(1 + \sigma_{mm} g_i), \qquad i = 1, 2, 3, \cdots$$
$$= n + \sigma_{cm} G + n\sigma_{mm} g_i + \sigma_{cm} G \sigma_{mm} g_i$$

Nominal + **Chip-mean variation** **+ ACV +** **Un-expected coupling term**

Correct treatment: $p_i = n + \sigma_{cm} G + n\sigma_{mm} g_i, \quad i = 1, 2, 3, \cdots.$

Element 4.

- **C. Enabling MC simulations of ACV at a chip-mean corner**
  - **Designers often need to ensure the functionality of their circuits at the worst chip-mean corner**

$$p_i = n + \sigma_{cm} s + \sigma_{acv} g_i, \quad i = 1, 2, 3, \cdots$$

s **–** a skewing parameter

g_i **–** stochastic/random variables

Enabling circuit designers to performance a smaller number of MC runs at the chip-mean corner to obtain a larger sample of ACV (or mismatch).

Summary

- **Reviewed requirements for statistical SPICE models**
- **Presented our solutions to several elements in statistical SPICE models**
- **Our solutions establish a solid foundation for a good statistical SPICE models**
- **Many of our solutions have been used in IBM bulk Common Platform’s and SOI Alliance’s compact models**

Thank S. Khandelwal (at IBM India) for presenting this talk for us at 2009 Nanotech/WCM