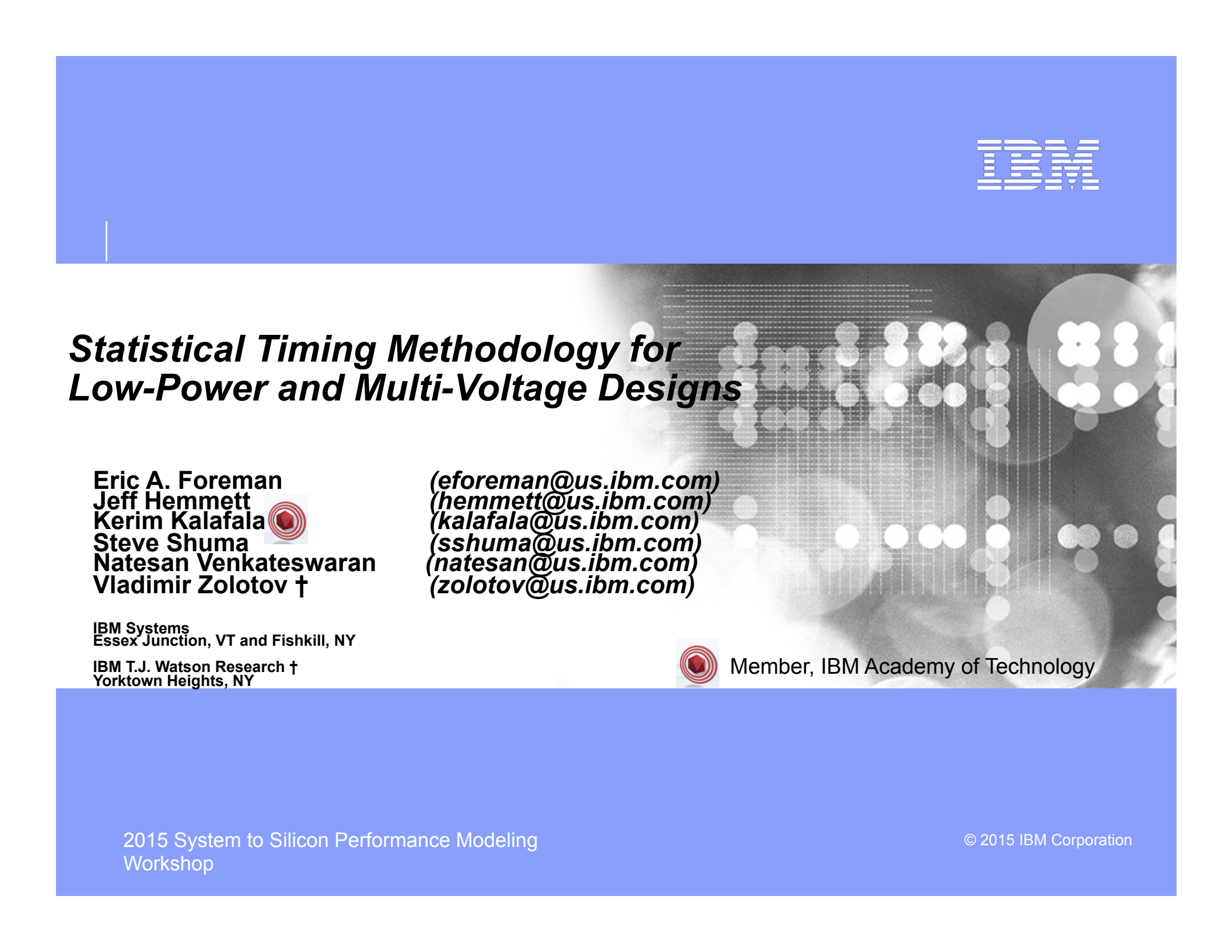


Statistical Timing Methodology for Low-Power and Multi-Voltage Designs

Eric A. Foreman *(eforeman@us.ibm.com)*
Jeff Hemmett *(hemmett@us.ibm.com)*
Kerim Kalafala *(kalafala@us.ibm.com)*
Steve Shuma *(sshuma@us.ibm.com)*
Natesan Venkateswaran *(natesan@us.ibm.com)*
Vladimir Zolotov † *(zolotov@us.ibm.com)*

IBM Systems
Essex Junction, VT and Fishkill, NY

IBM T.J. Watson Research †
Yorktown Heights, NY

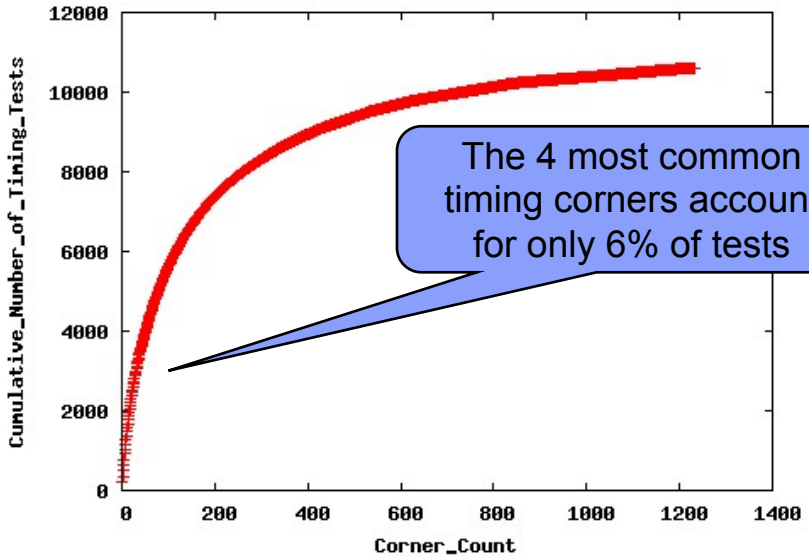
Member, IBM Academy of Technology

2015 System to Silicon Performance Modeling Workshop

Introduction / Outline

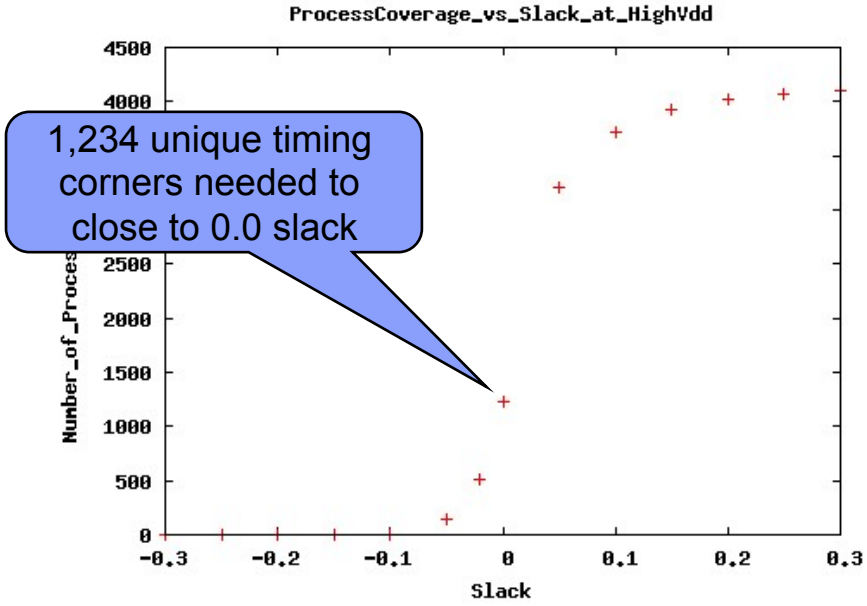
- Motivation
- Statistical Timing Overview
- Reliability Modeling
- Second Order Modeling
- Results
- Construction / Design Optimization
- Abstraction / Modeling Techniques
- Summary

Motivation for Statistical Timing - Avoiding the Exponential Trap of Corner-Based Timing



Number of unique corners at which at least one end point had its worst slack

Note: 4 most common timing corners account for 648 out of 10,616 negative slack tests



Statistical Timing - Canonical Model

- **The canonical model is a bi-linear delay distribution consisting of multiple sources of variation and cross terms with respect to a single base variable**
- **Every delay quantity still has an Early (smallest) and Late (largest) value represented by Early and Late canonical models**
- **The canonical model form is defined as:**

$$a_0 + a_1\Delta X_1 + a_2\Delta X_2 + a_{1,2}\Delta X_1\Delta X_2 \cdots + a_{n+1}\Delta R_a$$

Constant (nominal) value in the absence of variations

Sensitivities

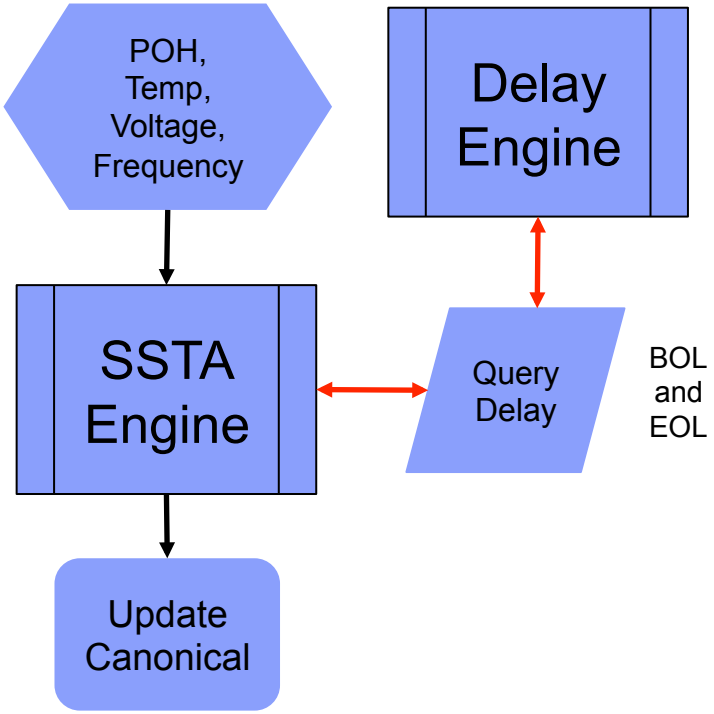
Global random variables; these are probability distributions

2nd order cross term

Independently random uncertainty

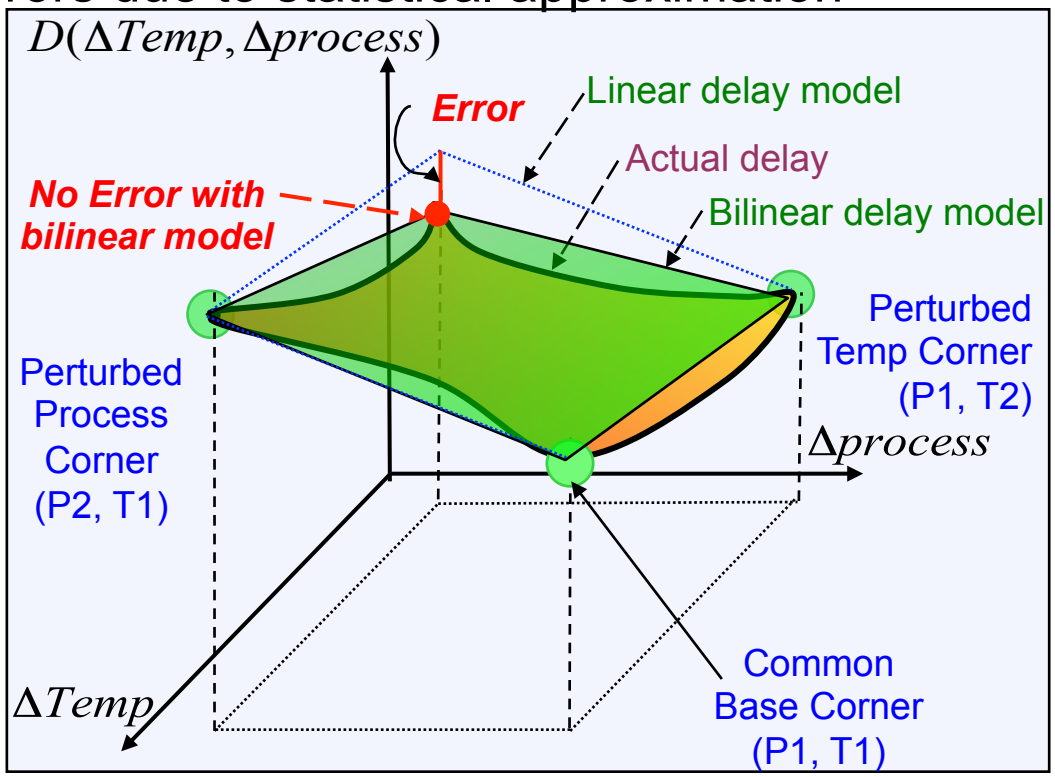
Product Reliability Modeling in SSTA

- **NBTI and Hot-e impacts transistor delay over time**
- **Delay Model Characterizes this**
  - Typically 4-10% at 3σ
- **SSTA incorporates source of variation using delay model**
- **SSTA computes variability to represent Beginning of Life and End of Life Conditions**
- **SSTA can tell you which condition is worse**



Bi-linear Modeling

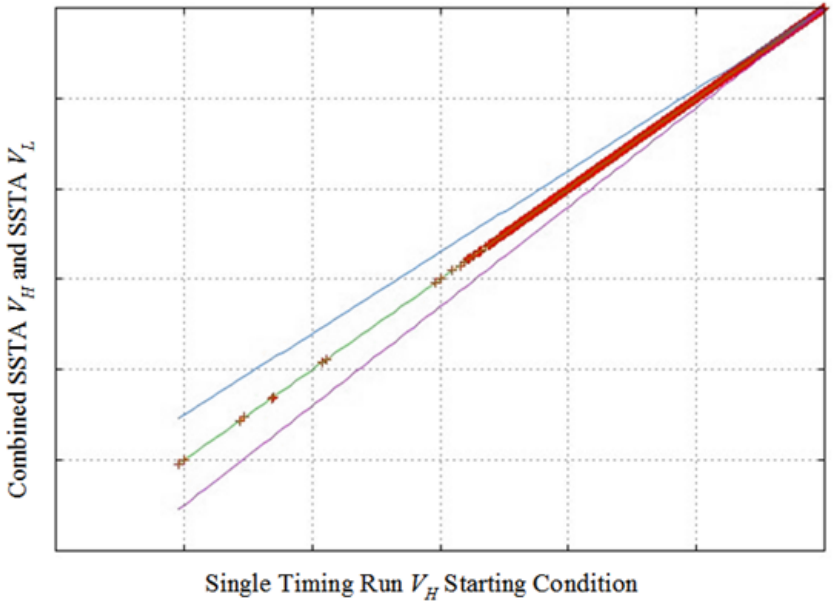
- Use of 2nd order cross terms to create a Single Timing Run (STR)
  - Bounds errors due to statistical approximation



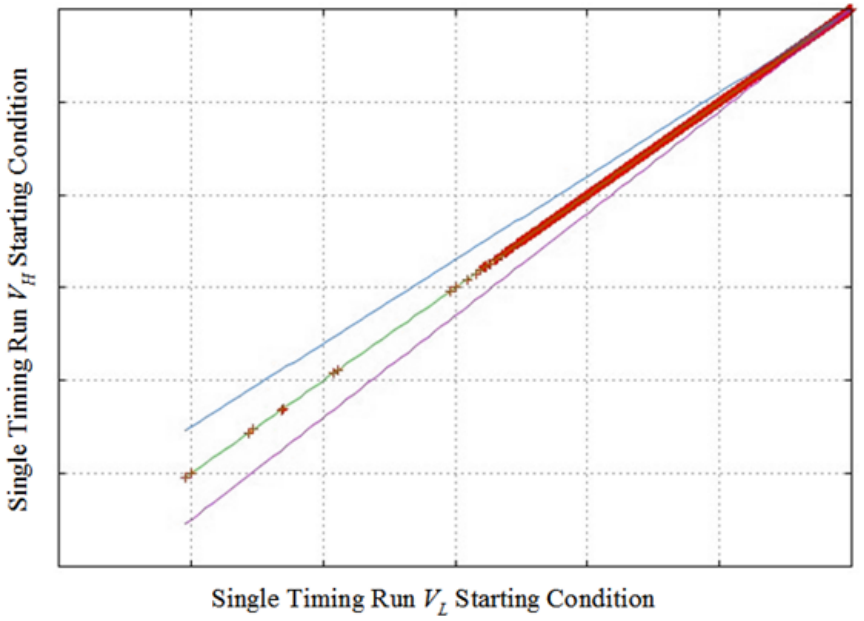
Actual delay shows variable ΔX is different at different Vdd values

The bi-linear model accounts for different ΔX across vdd space

Results



- X-axis: STR timing slacks with high voltage starting corner
- Y-axis: High voltage timing slacks and low voltage timing slacks from separate runs combined
- Conclusion: STR matches separate high and low voltage timing runs



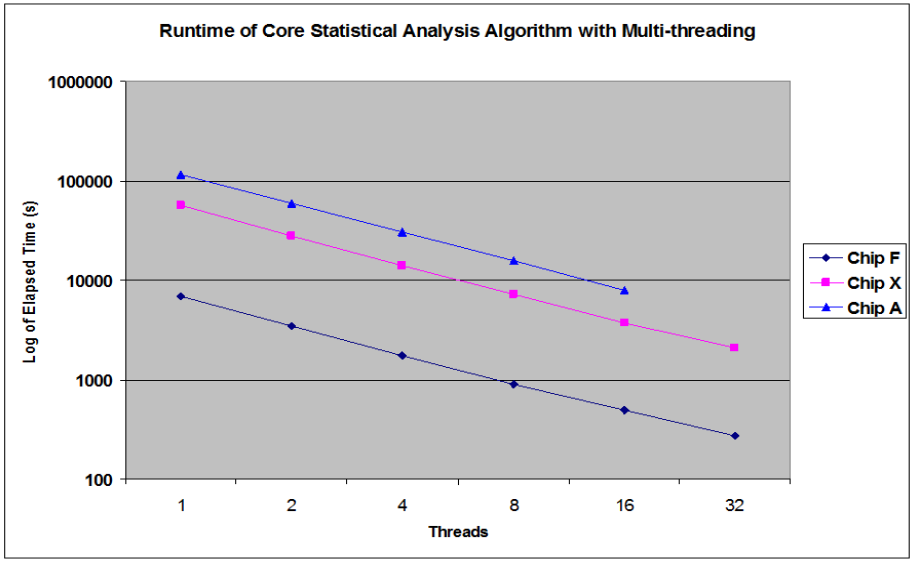
- X-axis: STR timing slacks with low voltage starting corner
- Y-axis: STR timing slacks with high voltage starting corner
- Conclusion: Starting corner selection gives same results

Runtime / Memory Comparison

- **Runtime**
  - Analysis is fully-threaded
  - Runtime decreases linearly with increase in # threads
  - For given X cpu machine:

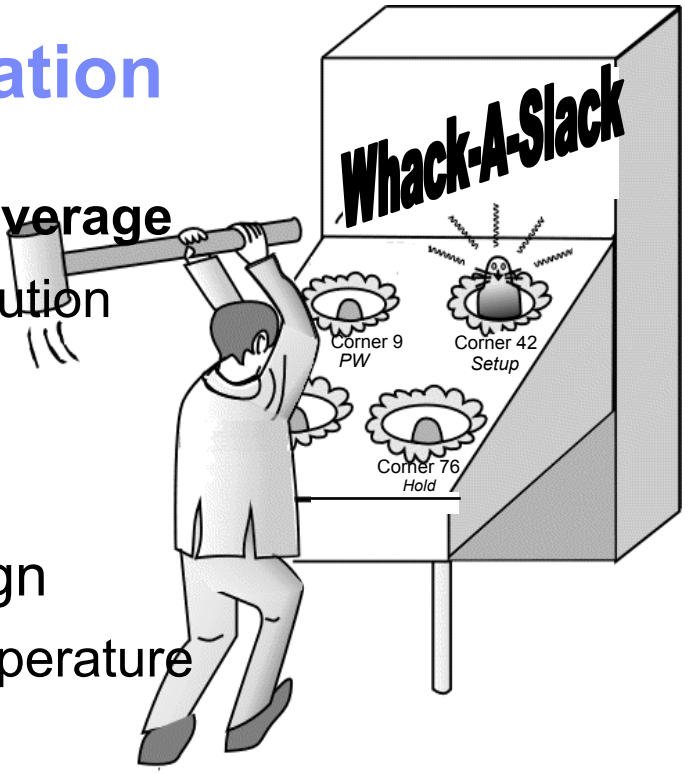
$$STR[Xcpu] \langle \max\left(TR_{VH}\left[\frac{X}{2}cpu\right], TR_{VL}\left[\frac{X}{2}cpu\right]\right)$$

  - Simulation results:
    - Show **16% Runtime Savings**



- **Memory**
  - STR loads timing models, parasitics, and netlist once
  - STR memory footprint is less than separate high and low voltage timing runs
  - Simulation results:
    - Show **30% Memory Savings**

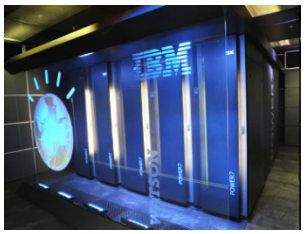
Construction Flows & Optimization

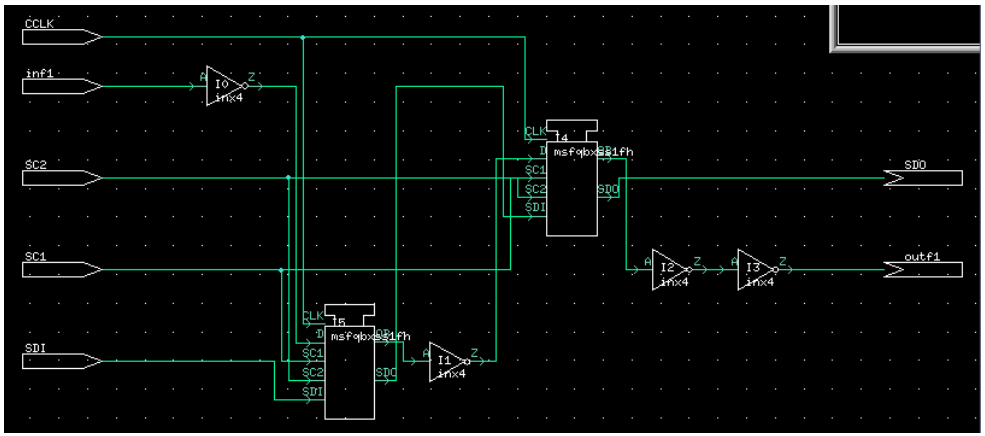


- **SSTA: Comprehensive process space coverage**
  - Analyzes entire space, returns limiting solution
    - Avoids 'whack-a-mole' failing test iterations
    - Avoids or reduces any required margining
  - SSTA flow can evolve along with design
    - Initially model Si-Process, Voltage, Temperature
    - Introduce metal parameters later in flow
    - Incremental parameter introduction can apply to other parms as well
  - Other methods for managing runtime
    - Many of the performance filters have variable accuracy
      - Use low accuracy/high performance early in flow (large number of large fails)
        - > Can use other more aggressive techniques early on (e.g. constant pincap)
      - Increase accuracy as design matures (small number of small fails)

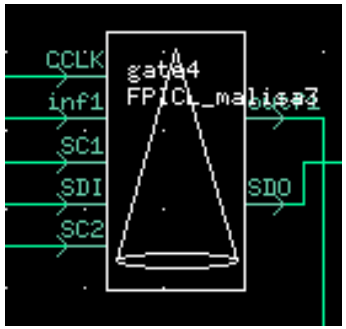
Statistical Abstraction: Enabling hierarchical timing with variability

- **Hierarchical designs: Partitioned (cores, units, macros)**
  (e.g. IBM POWER7-based Watson)

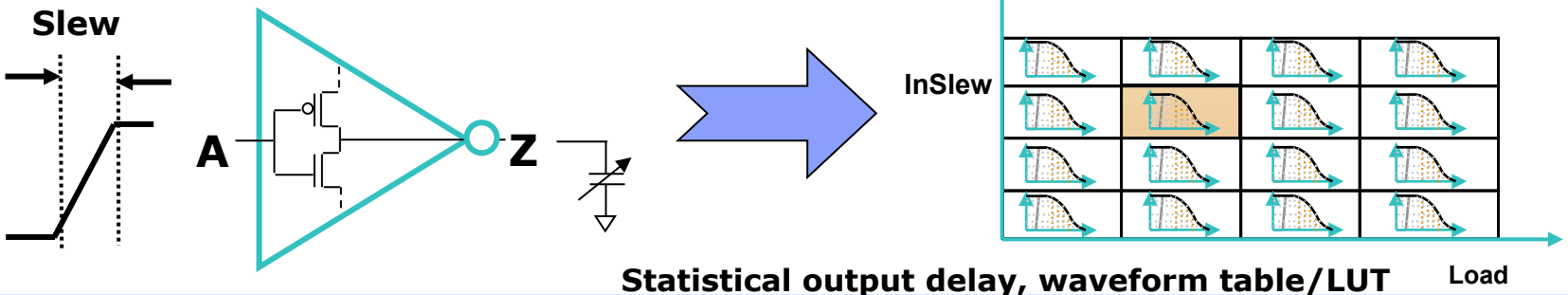




Abstraction



- **Statistical abstracts (macro-models) of partitions used at parent level**
  - Creates Look-Up-Table (LUT) for each arc's ***statistical***-delay and -output-waveform as function of *input-waveform's slew* and/or arc *capacitive-loading*



Summary

- ✓ Statistical timing allows us to cover an exponential # process corners in a single analysis framework
- ✓ 2nd order cross-terms are supported
- ✓ Accuracy has been validated against exhaustive corner based timing
- ✓ SSTA is fully threadable and amenable to incremental optimization flows