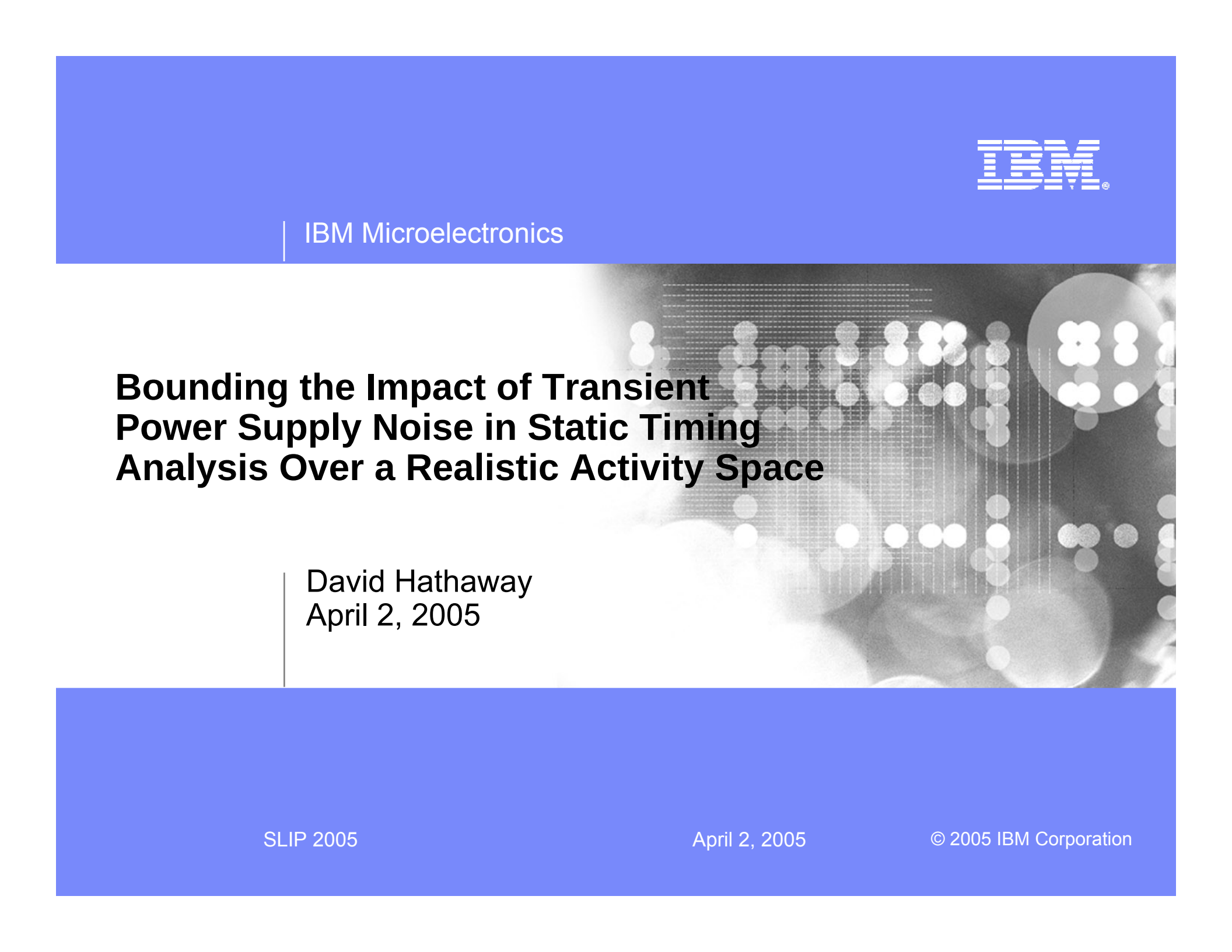


IBM Microelectronics

Bounding the Impact of Transient Power Supply Noise in Static Timing Analysis Over a Realistic Activity Space

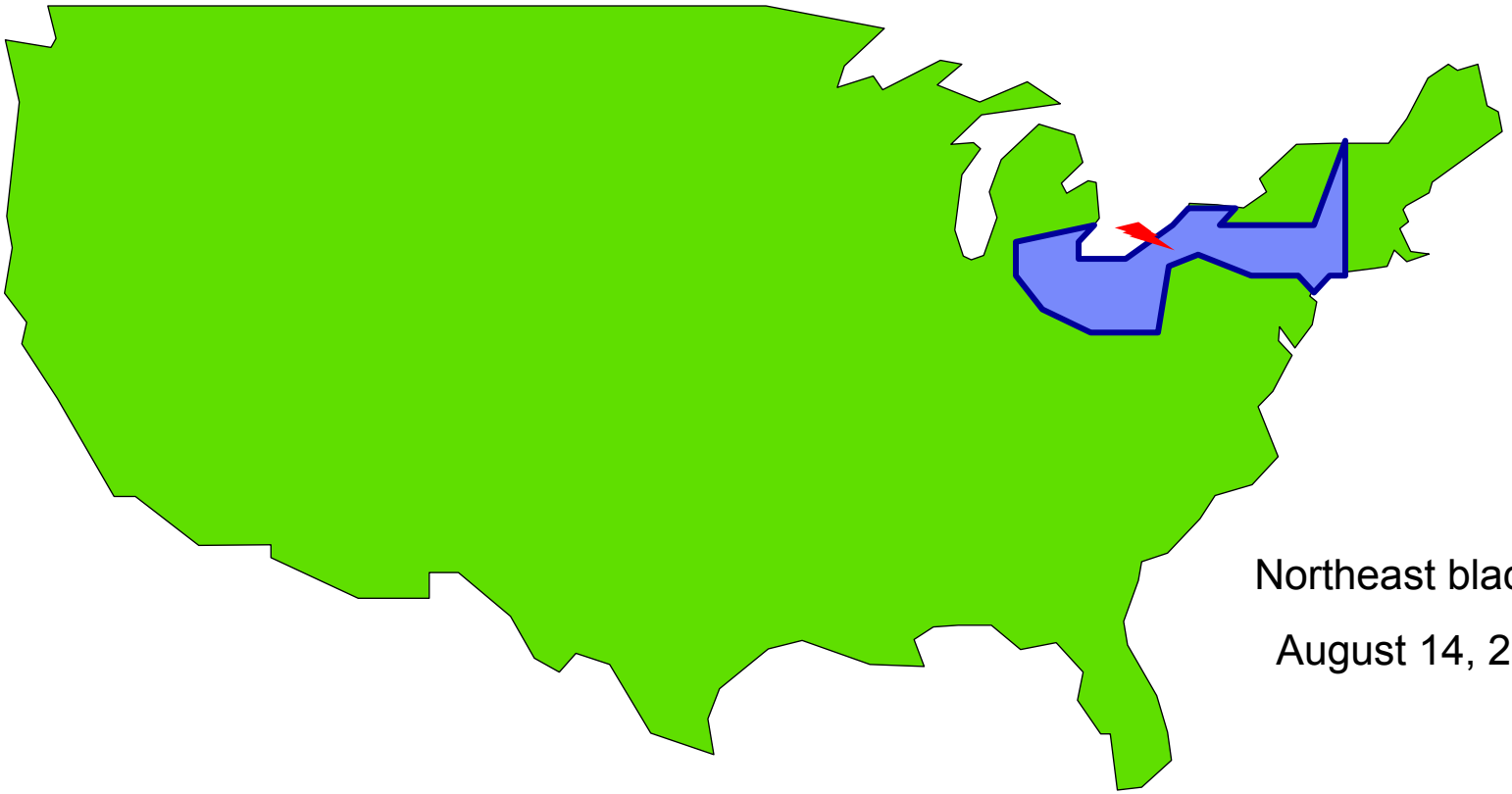
David Hathaway
April 2, 2005

Outline

- **Background and problem description**
- **Prior work**
- **This work**
  - Modeling realizable patterns of activity
  - Determining peak voltage variation
  - Determining bounding timing conditions
  - Applications to design planning
- **Remaining issues**

Power supply networks as System-Level Interconnect

- **Power supply networks do communicate information**
  - Power demand in one place affects voltages in others

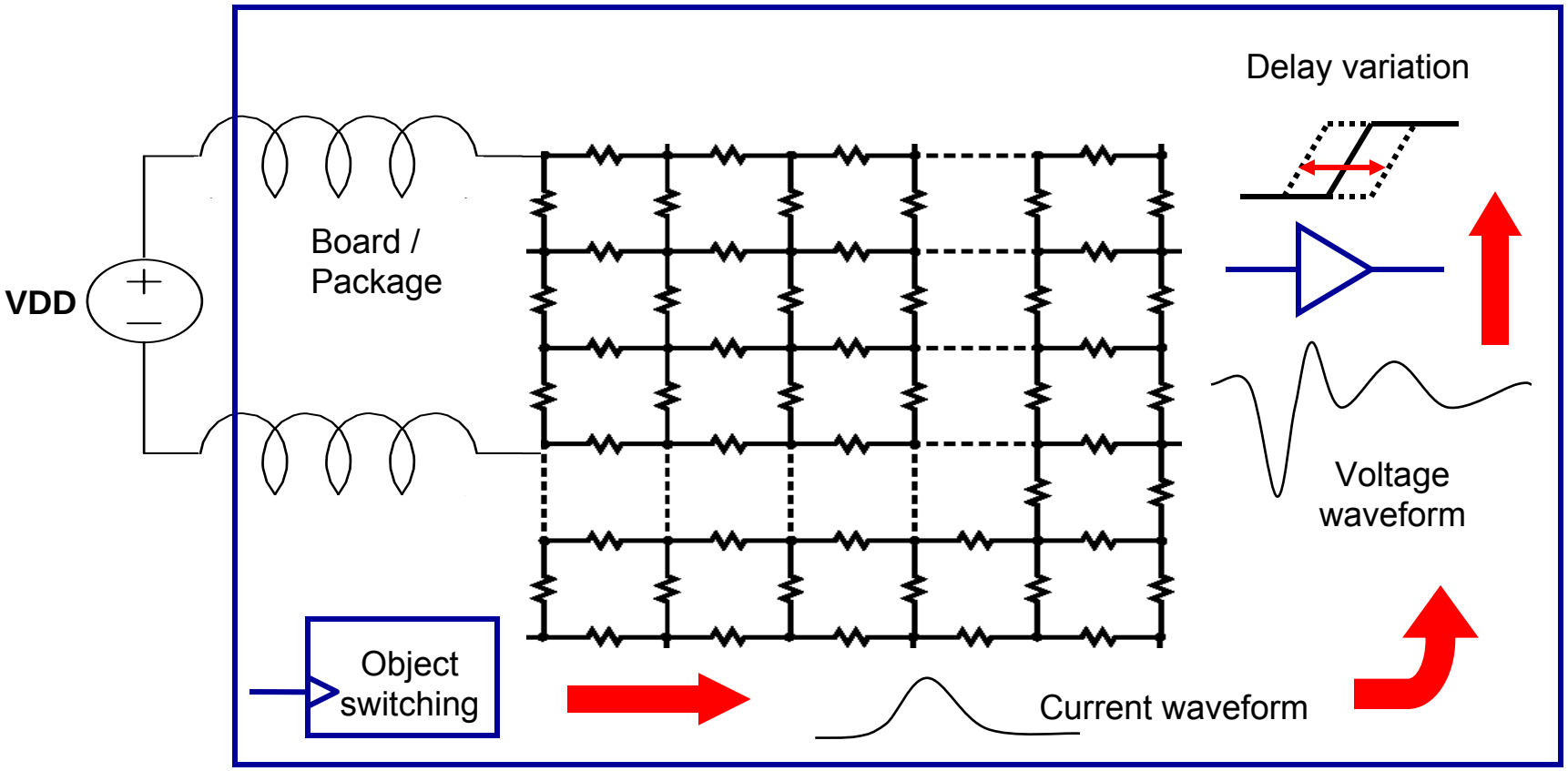


Northeast blackout

August 14, 2003

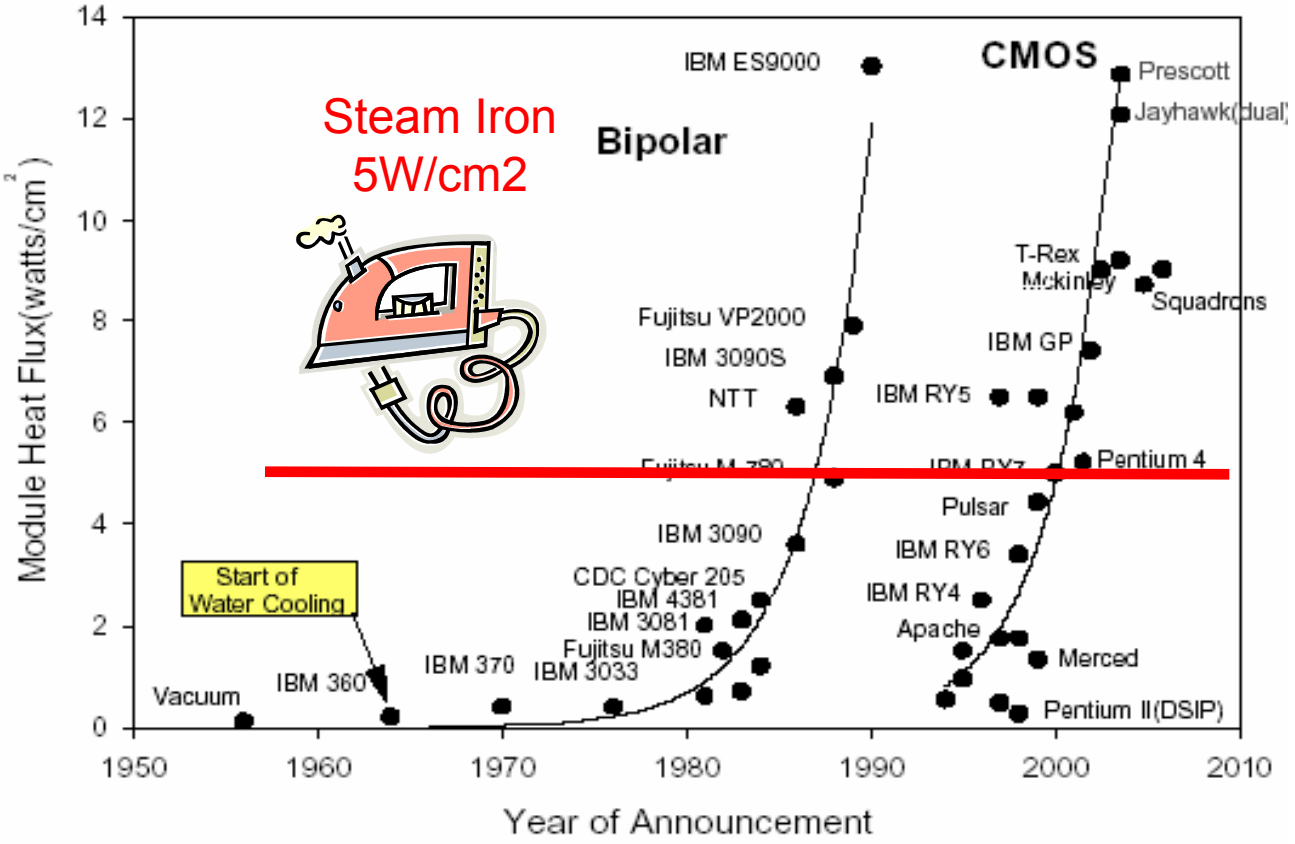
Problem addressed

- **Determine maximum impact of transient power supply noise on chip timing**



Why is this becoming more important?

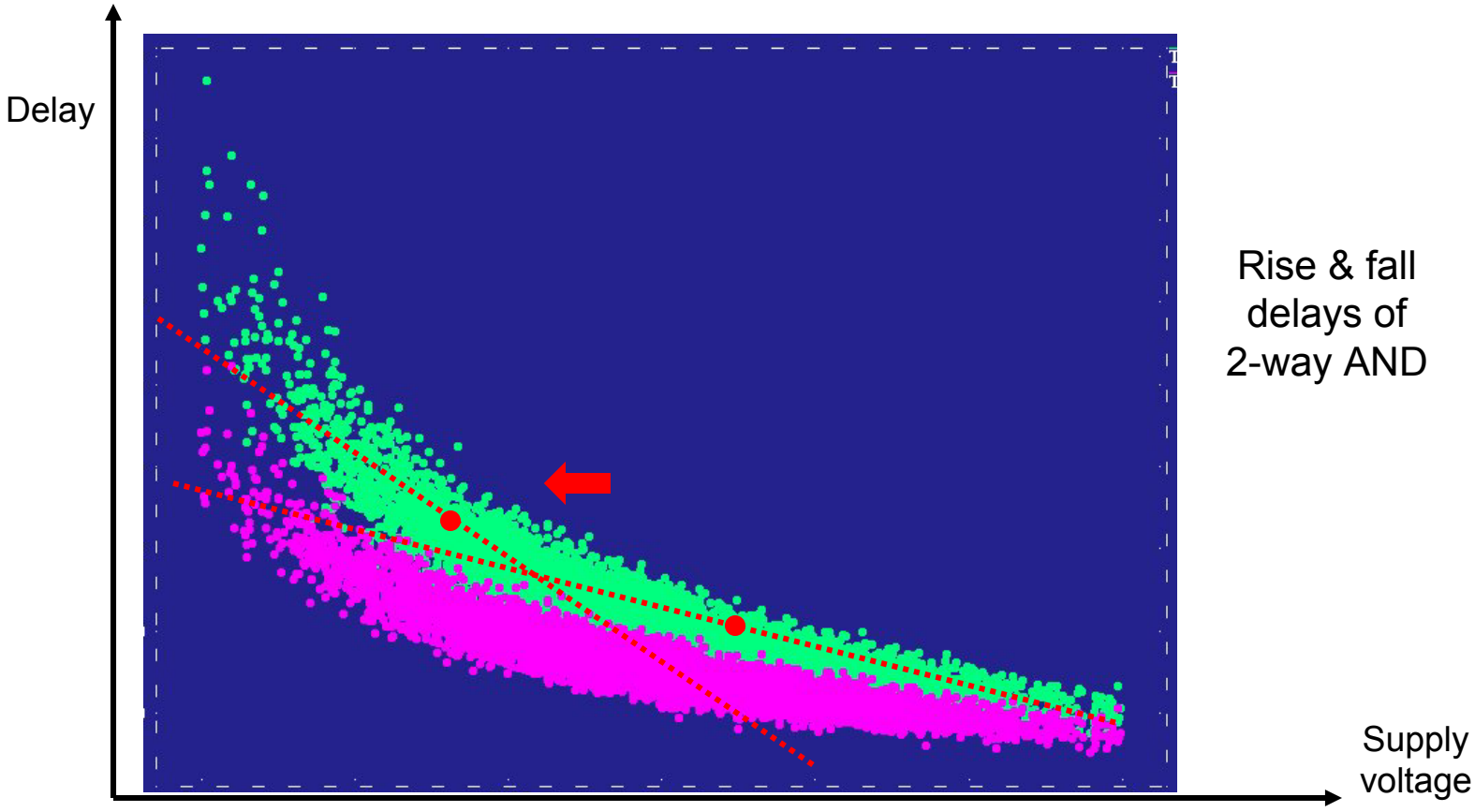
- **Power density is increasing (again)**



Why Hot Chips Are no Longer “COOL” - Ray Bryant

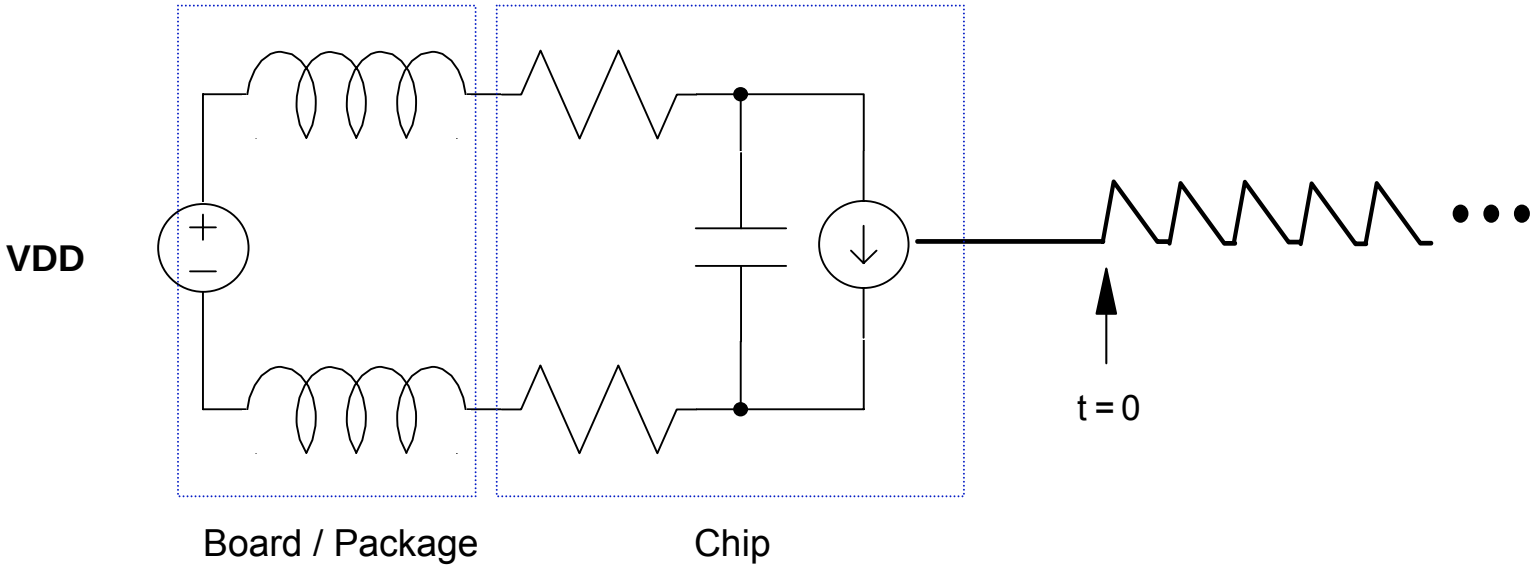
Why is this becoming more important?

- **Delay sensitivity to supply voltage is increasing**
  - Due to voltage reductions needed to contain power density



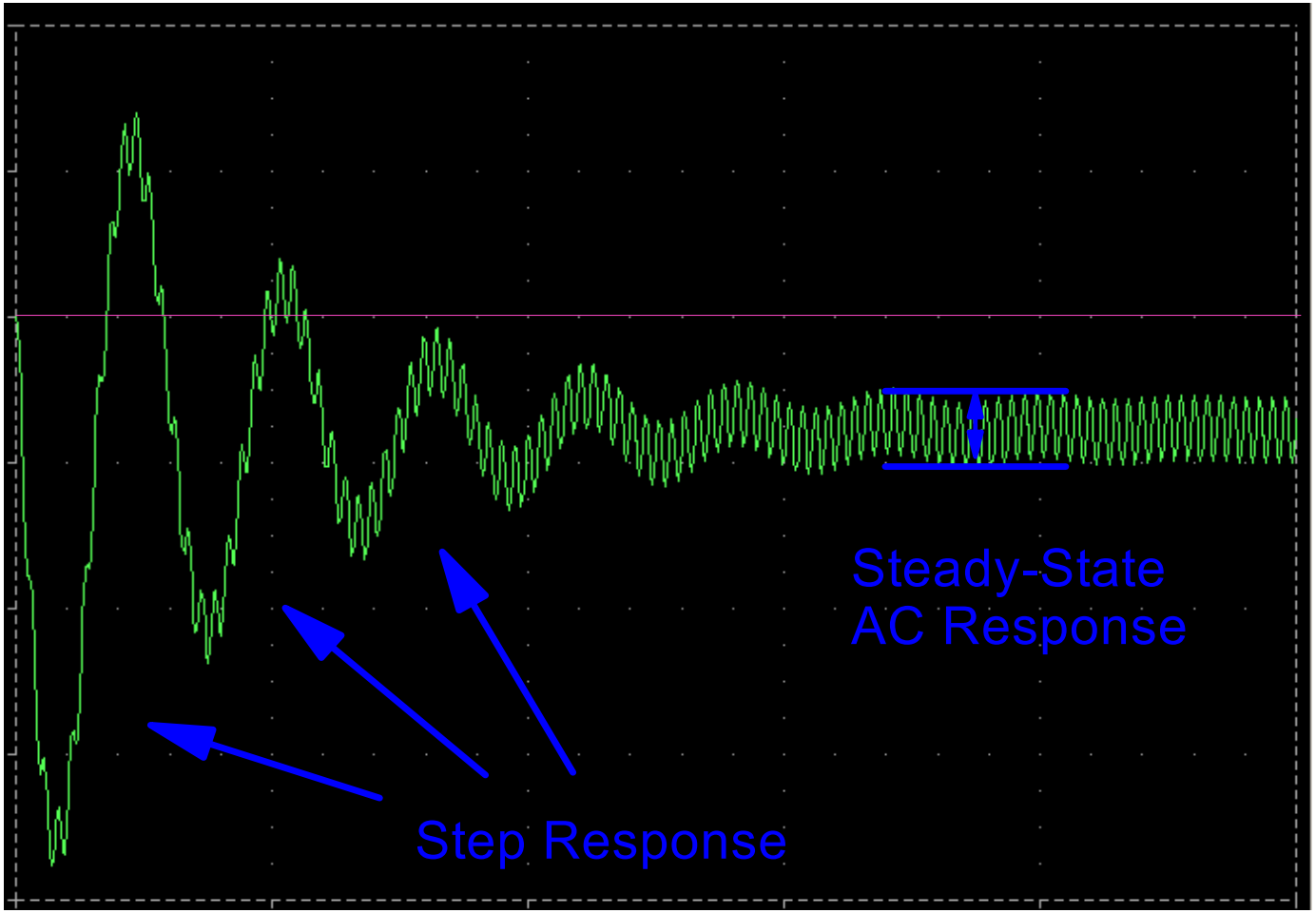
Power Supply Noise – Voltage Response

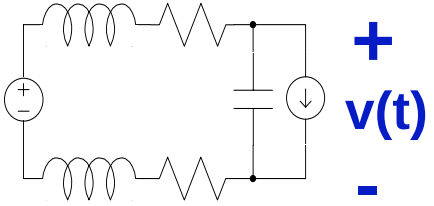
A trivial network model demonstrates key noise characteristics:

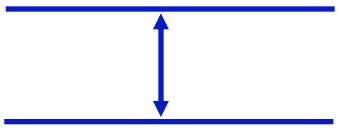


What is the "circuit's view" voltage response to the switching current signature illustrated above?

Voltage Response At Circuit Terminals



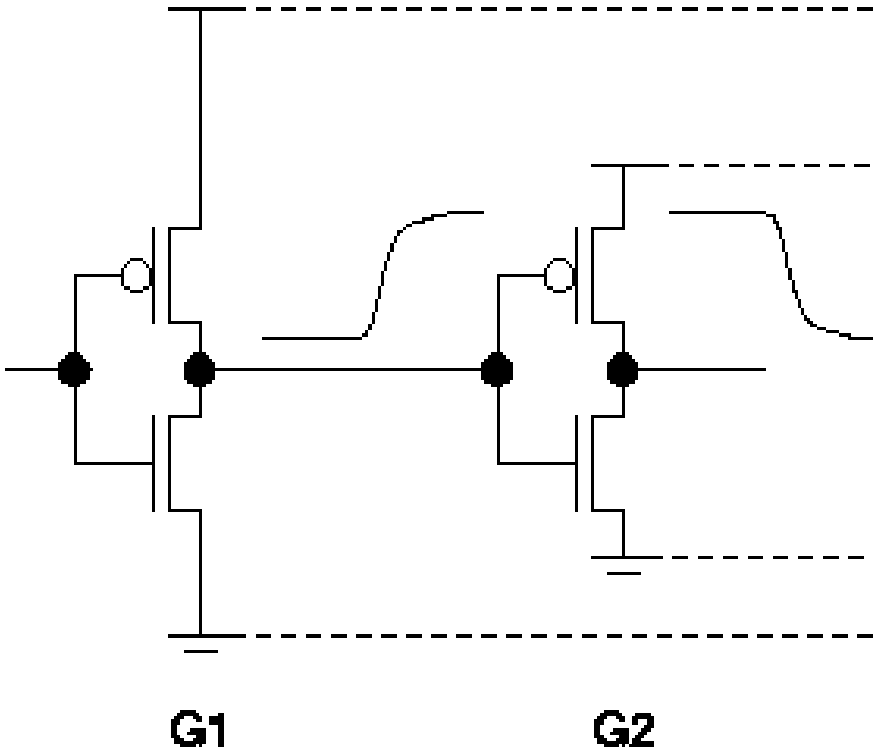




Static IR Drop

Delay dependence on voltage

- **Traditional models use a single supply voltage per gate**
  - Reality is more complicated

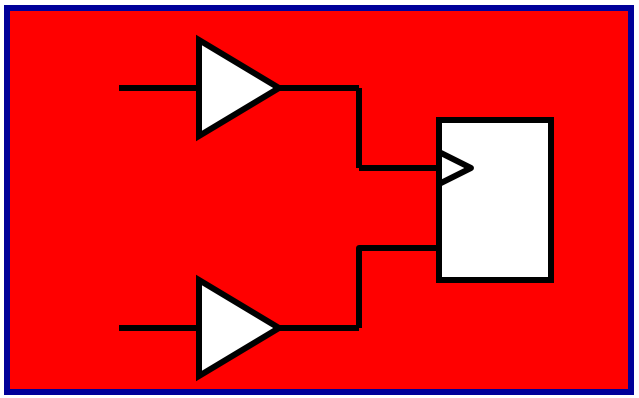


Drive (& hence delay) of gate G2 is a function of Vdd & Gnd of both G1 and G2

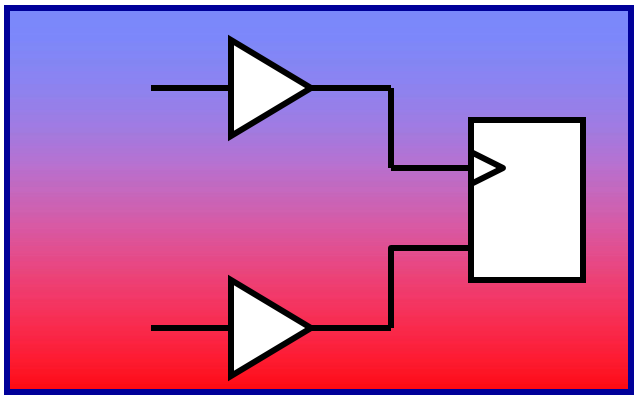
Most power supply noise is differential (Vdd drops as Gnd rises)

Why is it hard to find the worst timing impact?

- **Naïve approach**
  - Determine worst voltage at each node (max Gnd, min Vdd)
  - Time circuit with these voltages
- **Problems**
  - Timing test compare early/late clock/data
  - Worst slack (for setup test) when data is slow *relative to clock*



Max voltage drop
Max delay



Max voltage drop gradient
Worst slack

Outline

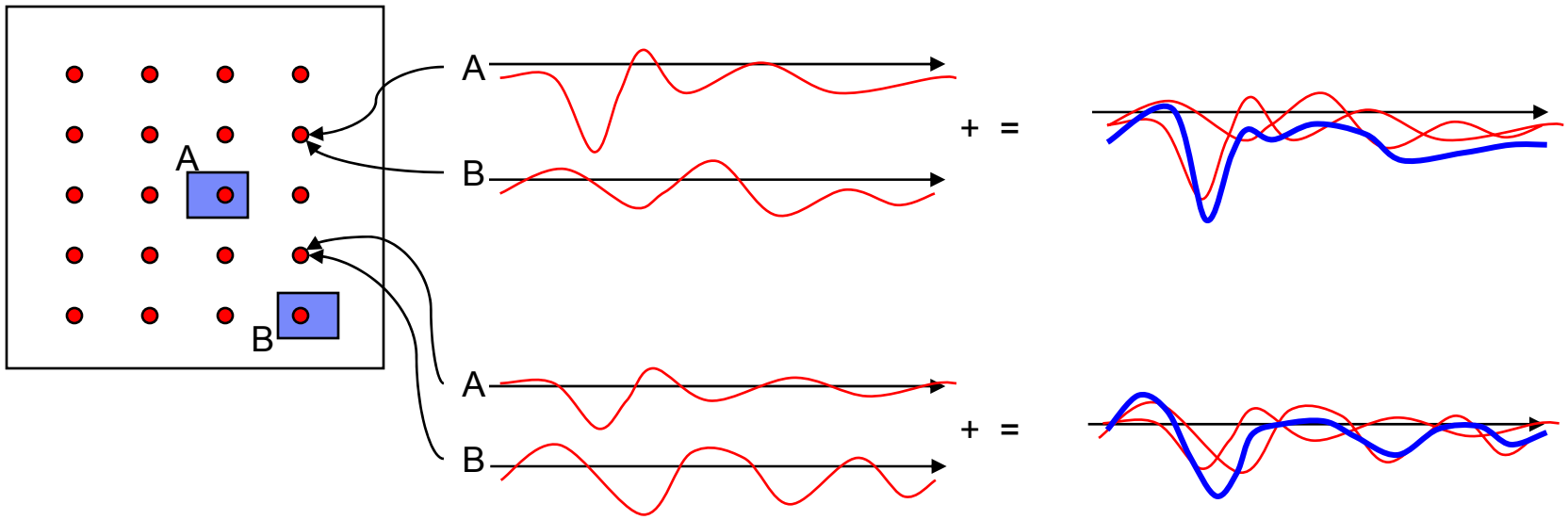
- **Background and problem description**
- **Prior work**
- **This work**
  - Modeling realizable patterns of activity
  - Determining peak voltage variation
  - Determining bounding timing conditions
  - Applications to design planning
- **Summary**

Use superposition to determine worst timing

- ***Vectorless Analysis of Supply Noise Induced Delay Variation*, S. Pant, D. Blaauw, et. al., U. Michigan, ICCAD 2003**
  - Bound current demand at each power network node
  - Simulate current impulse applied at each power network node
  - Model delay linearly with voltage
  - Use delay model & power network response to model path delay as function of current at each node in each cycle
  - Use linear optimizer to determine current profiles which give worst path delay
  - Use spatial and temporal superposition of voltage waveforms

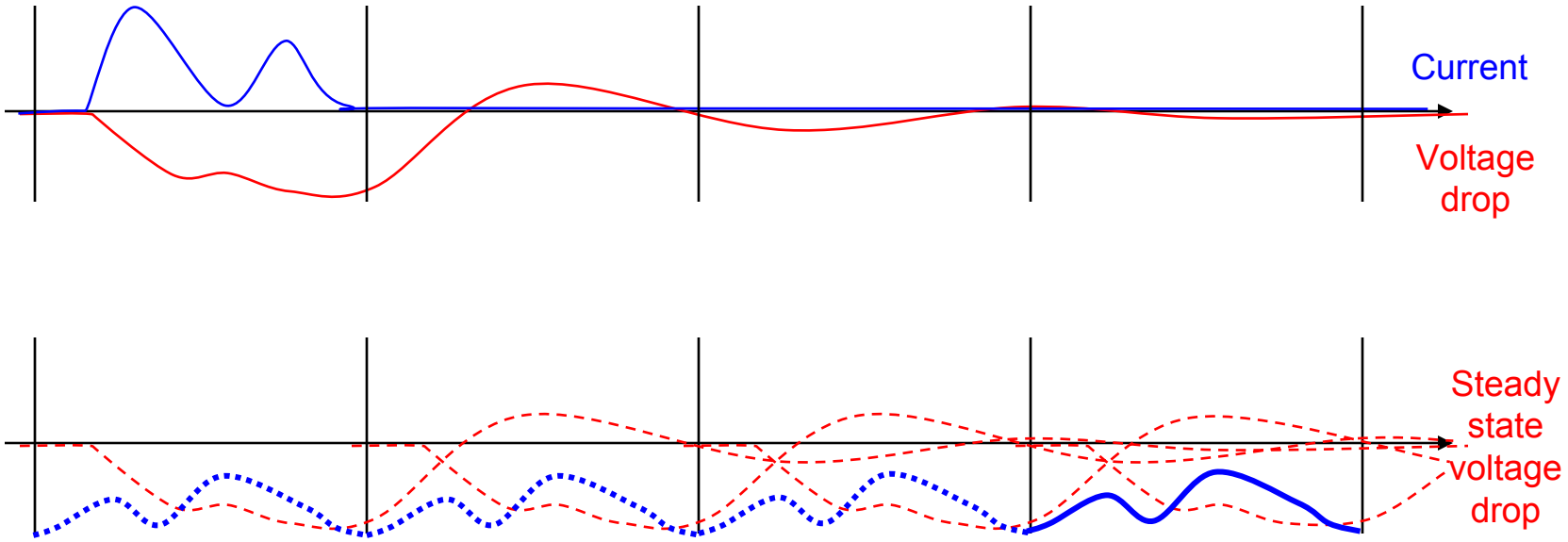
Spatial superpositioning

- **Compute voltage waveforms for different aggressors**
- **Add selected waveforms with no time shifting**

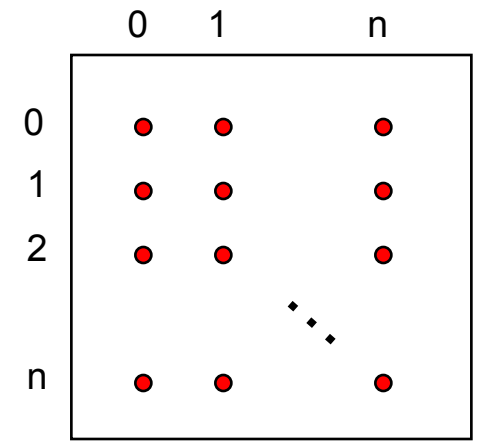


Temporal superpositioning

- **Compute voltage waveforms for one cycle of each aggressor (simulate until transients die out)**
- **Add time shifted copies of waveform to get impact of operation over many cycles**

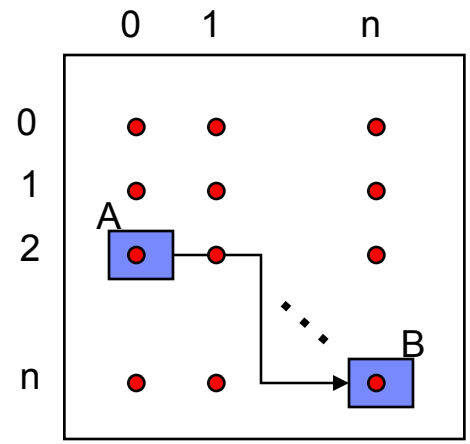


Use superposition to determine worst timing

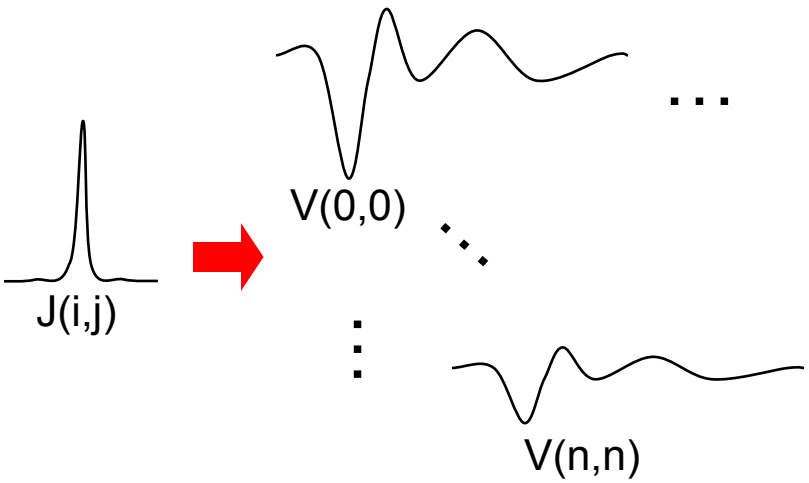


$$D = d_0 + \frac{\partial d}{\partial v}\Delta v$$

$$= d_0 + \sum_{ij} \frac{\partial d}{\partial v}\frac{\partial v}{\partial J_{ij}}\Delta J_{ij}$$



Σ



$$D_{path} = d_{0,path} + \sum_{i \in nodes}\left(\sum_{j \in cycles}\frac{\partial d_{path}}{\partial J_{ij}}\Delta J_{ij}\right)$$

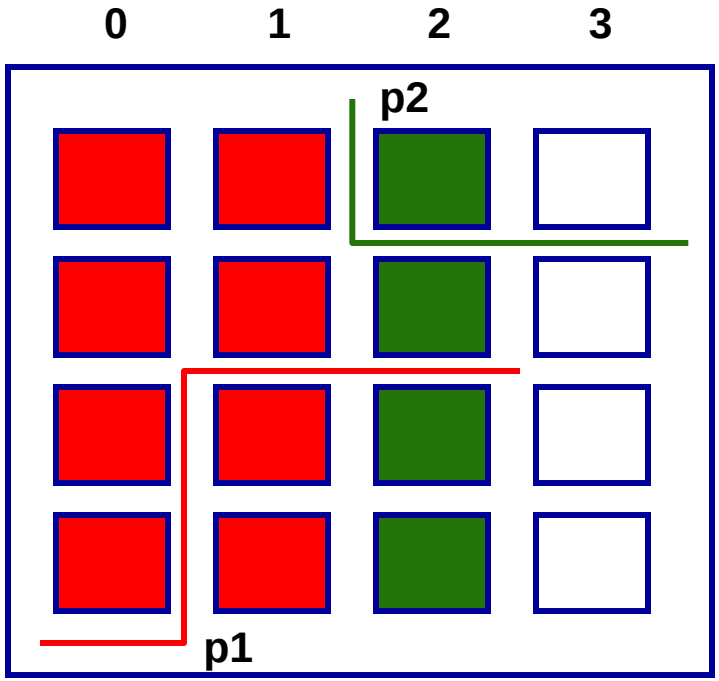
Find ΔJ_{ij} to maximize D_{path}

Limitations of prior work

- **Cannot easily restrict analysis to *realizable* activities**
  - Allows only linear constraints between current weights
- **Path-oriented analysis**
  - Subject to exponential path enumeration issues
- **Assumes constant voltage within cycle**
  - Loses high frequency variation

Constraints on realizable activities

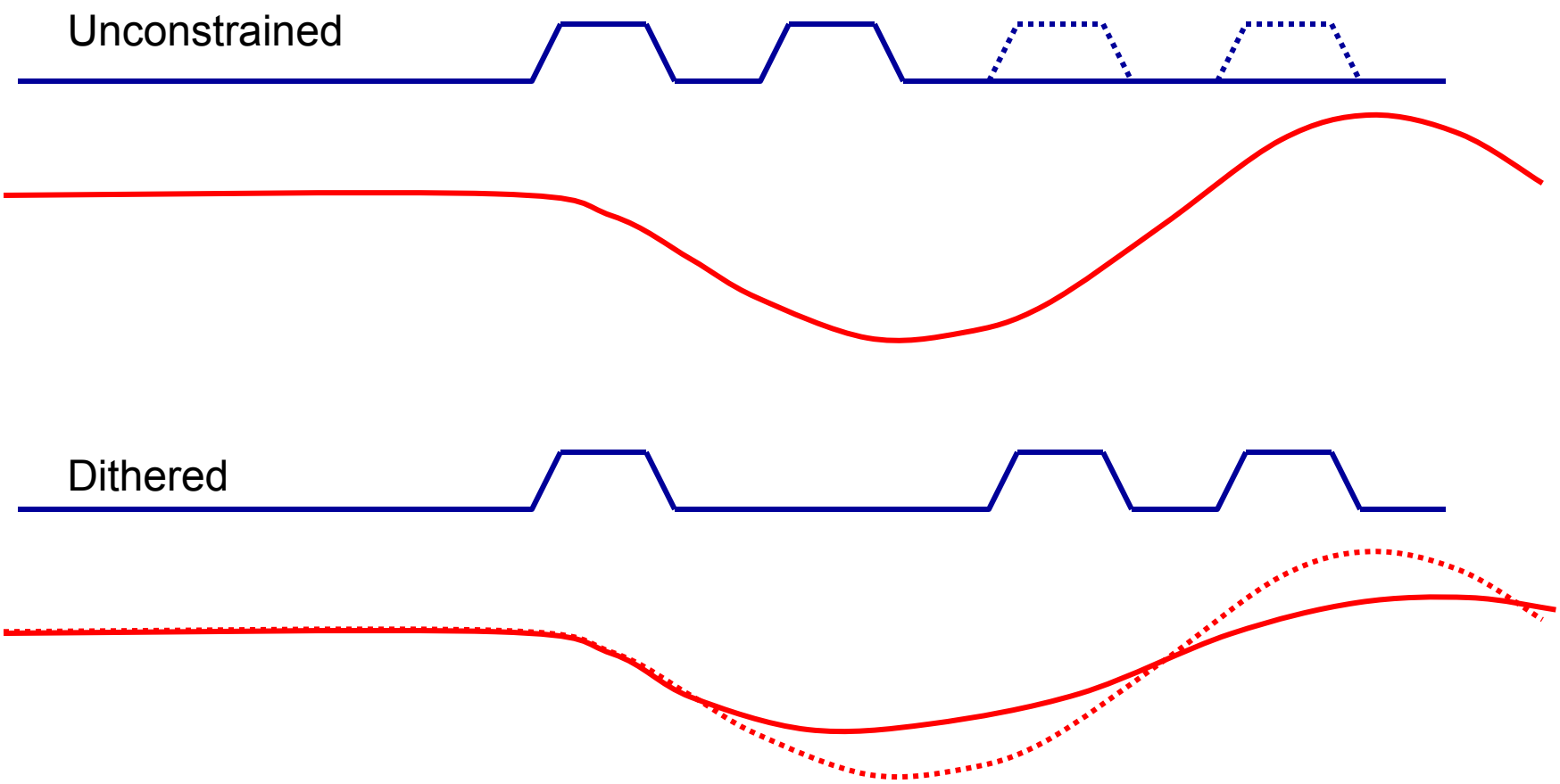
- **Example – either banks 0 & 1 switch *or* bank 2 switches *or* bank 3 switches**



- **Banks with worst effect differ for paths p1 and p2**
- **Cannot constraint total switching**
  - S(0) + S(1) > S(2) or S(3) alone
- **Cannot constrain relation between banks**
  - S(0), S(1) > S(2) for p1
  - S(2) > S(0), S(1) for p1
- **Selection of worst banks is not a linear problem**

Constraints on realizable activities

- **Example – clock dithering to limit noise from clock gating**



Outline

- **Background and problem description**
- **Prior work**
- **This work**
  - Modeling realizable patterns of activity
  - Determining peak voltage variation
  - Determining bounding timing conditions
  - Applications to design planning
- **Summary**

Overview of our approach

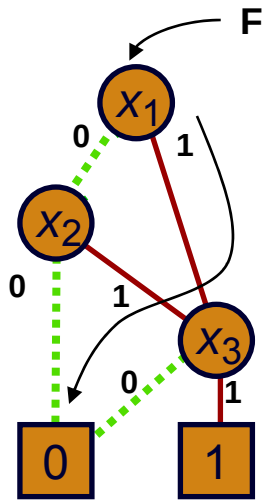
- **Identify "objects of interest" (OOIs)**
  - Large or high power cores of SOC
  - Regions of random logic
- **Simulate voltage drops due to each OOI**
- **Determine allowable patterns / sequences of activity for OOIs**
  - Modeled as BDDs
- **Determine max / min v(t) for each node**
- **Use min / max v to screen, find critical subnetwork**
- **Use block-based statistical timing with OOI activities as variables to refine timing**
- **Determine worst slack for each timing test within allowable sequences**
- **Reanalyze paths with nonlinear delay dependencies to validate / refine slacks**

Simulating OOI currents

- **Use fast linear power grid simulator**
  - Uses fixed time step, explicit matrix inversion
    - Initial overhead
  - Allows very fast simulation of many waveforms
  - Changing OOI currents, location of application is very fast
  - Changing power grid (or adding decoupling caps) is slower

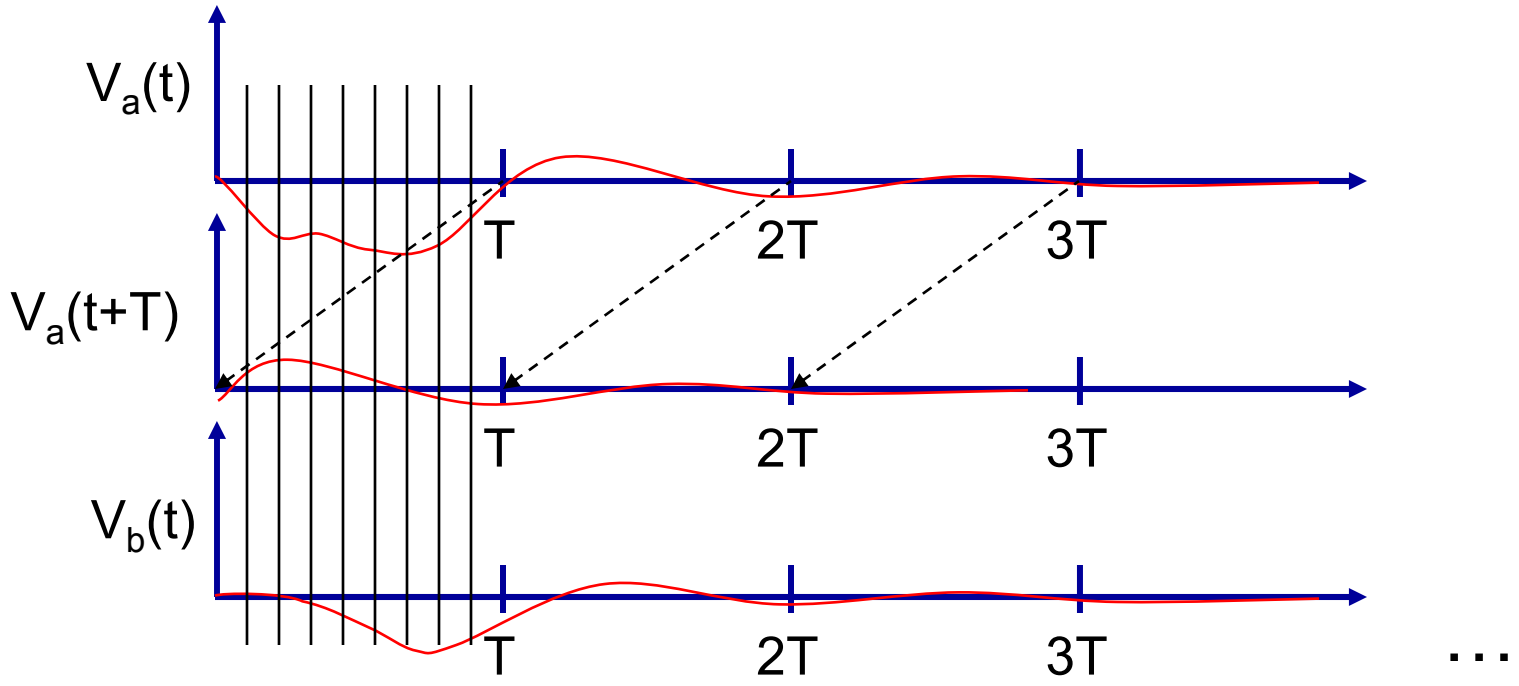
Modeling allowable conditions

- **Determine set of conditions which must be met**
  - Model as Boolean functions of OOI activity in different cycles
  - Examples
    - No more than k of n memory banks switch in one cycle
    - Clock cannot have X cycles off followed by Y cycles on
- **Represent AND of all conditions as BDD**
  - One BDD for all chip constraints
  - Variables are activity of OOIs in particular cycles
  - Subsequent steps depend linearly on BDD size
  - … so aggressively reorder for minimum size



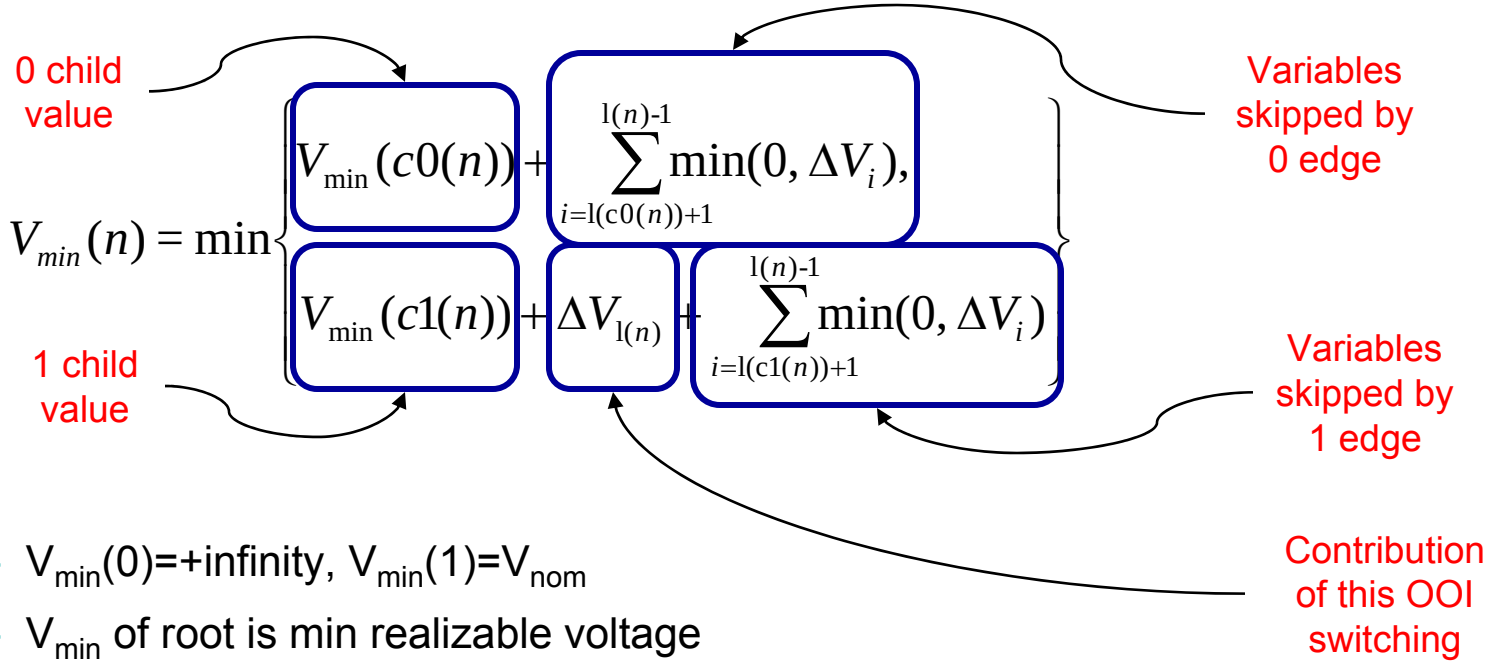
Finding extreme voltages

- **DFS traversal of constraint BDD**
  - One traversal per node, per alignment in cycle
  - Assign weights to BDD variables based on voltage of node at alignment time due to OOI switching in a particular cycle
  - Determine extreme (min/max) value at each node



Finding extreme voltages

- **For each node / alignment**
  - ΔV_i = delta voltage when OOI_i is active
  - Min value $V_{min}(n)$ at BDD node n with variable l(n), children c0(n), c1(n):



  - $V_{min}(0)$=+infinity, $V_{min}(1)=V_{nom}$
  - V_{min} of root is min realizable voltage
  - Similar form for max voltage

Finding extreme voltages

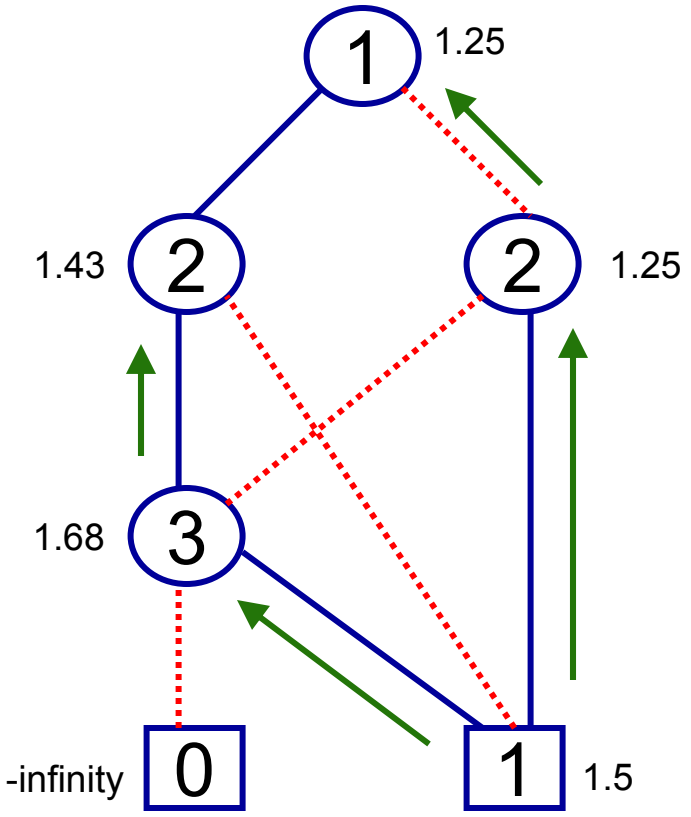
- **Example**

$V_1 = -0.1$

$V_2 = -0.25$

$V_3 = +0.18$

Node / offset 1



Finding extreme voltages

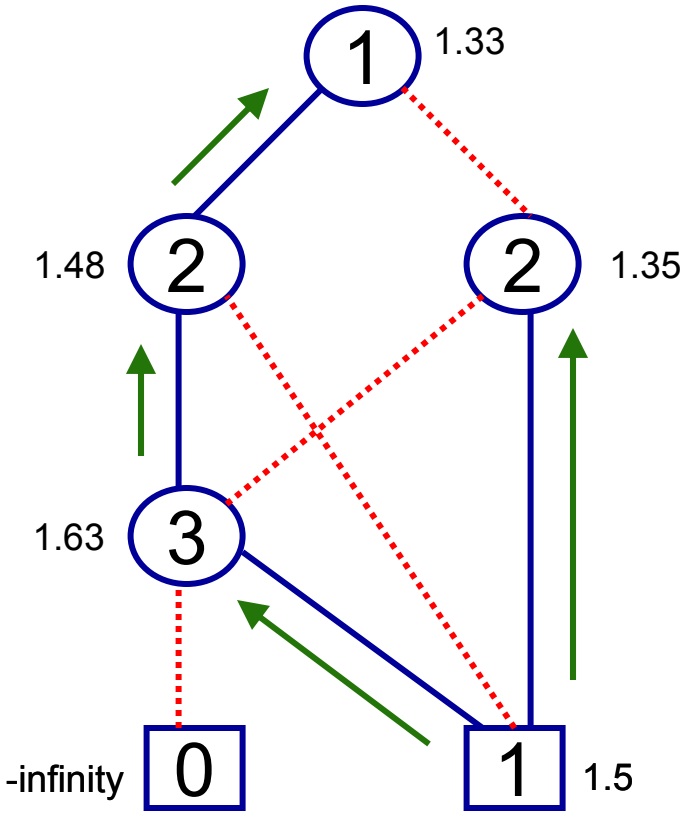
- **Example**

$V_1 = -0.15$

$V_2 = -0.15$

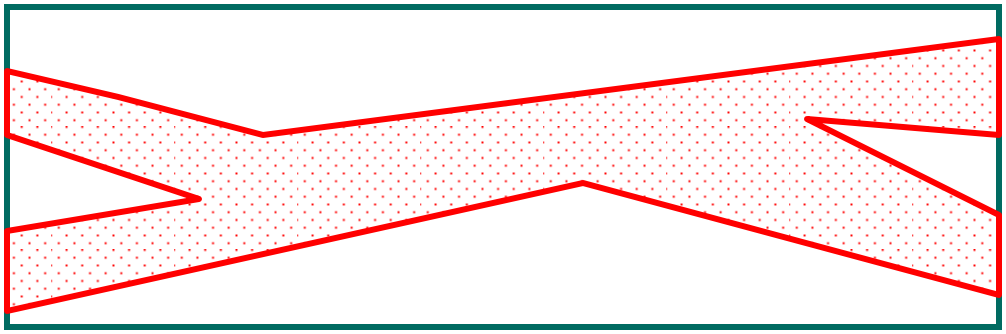
$V_3 = +0.13$

Node / offset 2



Application to timing

1. **Use min (late) / max (early) voltages to filter errors**



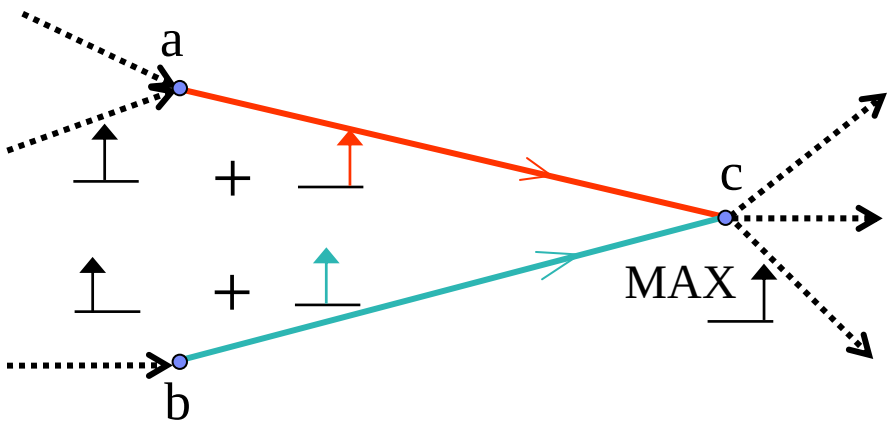
2. **Apply block-based statistical timing to remaining regions**
   - Delay / Arrival times / slacks are functions of OOI activity

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

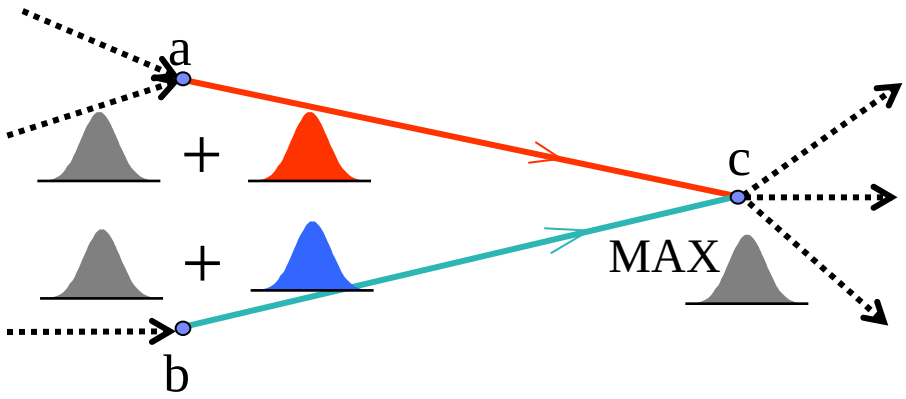
   - Max / min propagated statistically
   - Sample waveform voltages at mean arrival time
   - Result is slack equation for each test
     - Find worst slack by BDD traversal

Block-based statistical timing

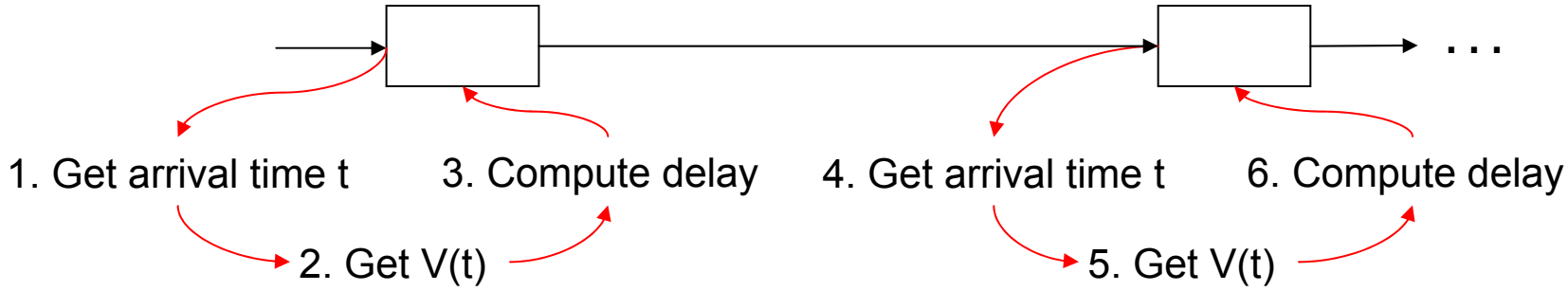
- **Deterministic**



- **Statistical**



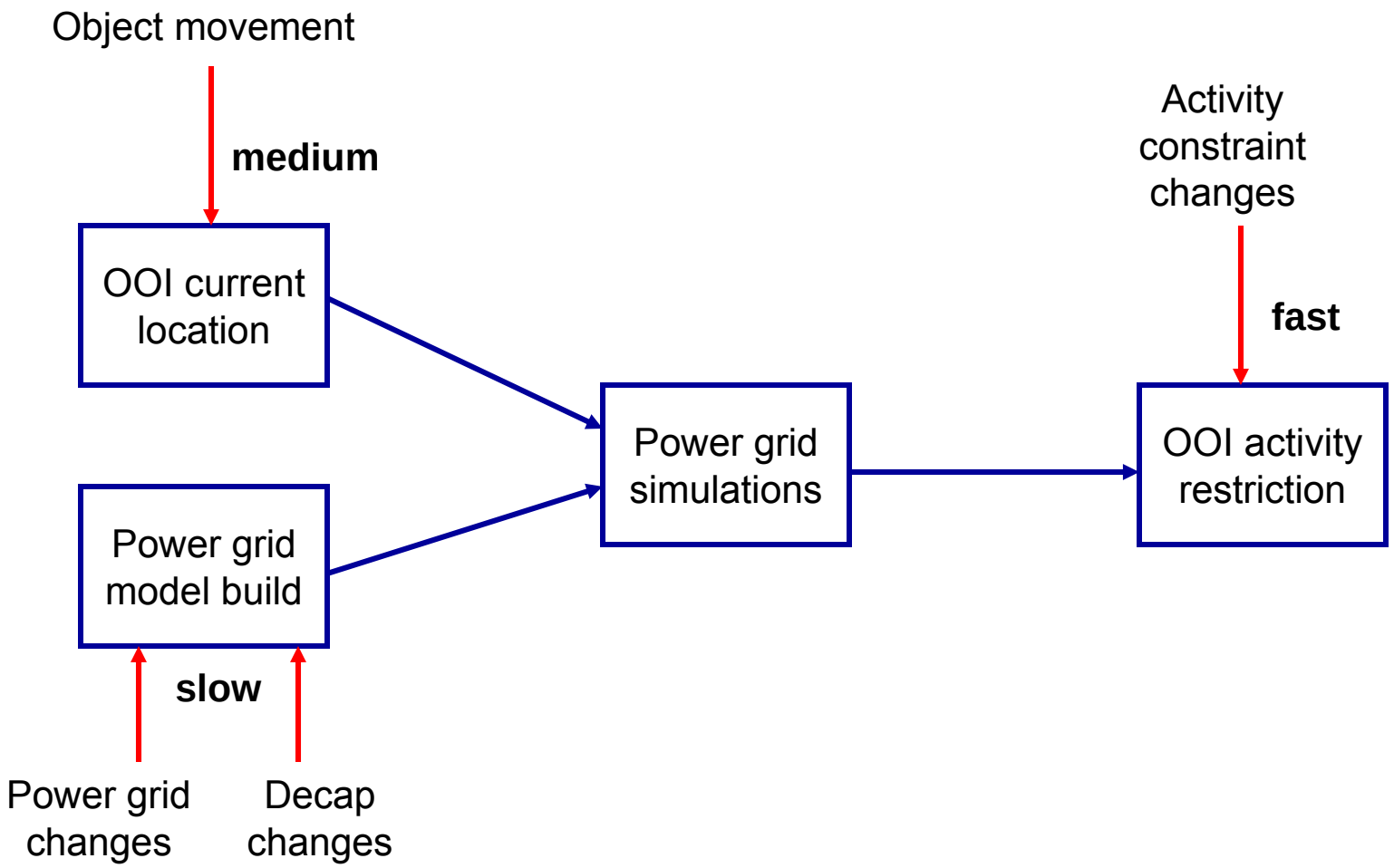
Waveform sampling for delay calculation



Applications to Design Planning

- **Get OOI locations from floorplan**
- **Determine extreme voltages**
  - Detailed paths not yet known, so can't apply to timing
  - Impose bounds on voltage
- **Modifications to fix errors**
  - Add decoupling caps
    - Only useful for short transient limit violations
  - Move OOIs
  - Impose restrictions on allowed activity patterns

Updating voltage extremes



(Some) remaining issues

- **Current depends on voltage**
  - Ignored by superposition approach
- **Block-based statistical timing may not apply well**
  - Need to validate predicted worst activity patterns
  - Fall-backs - use path-based approach
- **Picking the right set / granularity of OOIs**
- **Determining activity constraints**
  - Need links to system-level modeling

Acknowledgements

- **Thanks to many people in IBM working on these issues, and in particular:**
  - Ivan Wemple
  - Chandu Visweswariah
  - Sani Nassif
  - Doug Stout
  - Kerim Kalafala

