

Intrinsic Heterogeneity in Multicores Due to Process Variation and Core Aging

Josep Torrellas

Department of Computer Science
University of Illinois at Urbana-Champaign
<http://iacoma.cs.uiuc.edu>

Supported by NSF, Intel and IBM



Multiprocessor Architectures for Speculative Multithreading
Josep Torrellas, University of Illinois



Motivation

- Chips continue to integrate **more** and **lower-quality** transistors
- Process variation: **deviation** of transistor parameters from **nominal specifications**
- Impact on multicores:
 - **Spatial variation**: different cores on chip run at different frequencies and consume different power
 - **Temporal variation**: cores age depending on the load
- Result: homogeneous multicores are really **heterogeneous**
- This talk:
 - Understand these effects
 - How to exploit/minimize this heterogeneity

Executive Summary

- Spatial variation: chip has spatial localities
- Temporal variation: aging is exponential on V and T
- To exploit spatial variation in a multicore:
 - Variation-aware job scheduling and power management
- To minimize temporal variation:
 - Hide and slow down aging with load and voltage changes
- What will happen in the next few years:
 - Multicores will become more **dynamic** (sensors/actuators)
 - Homogeneous multicores can be turned **heterogeneous**
 - Designs will become more **aggressive** (less provision for worst-case use)

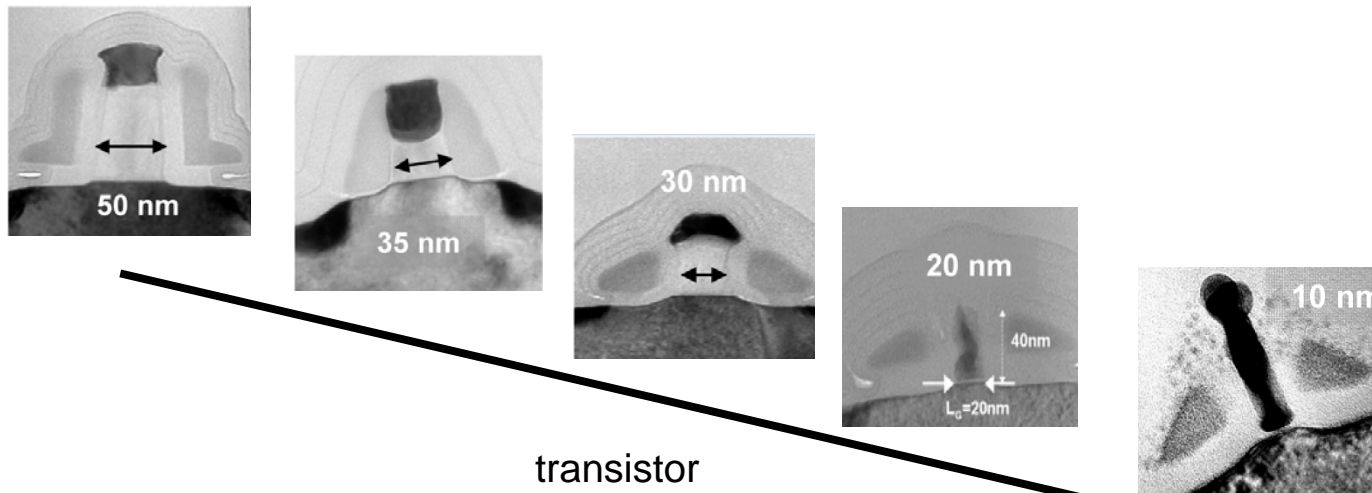
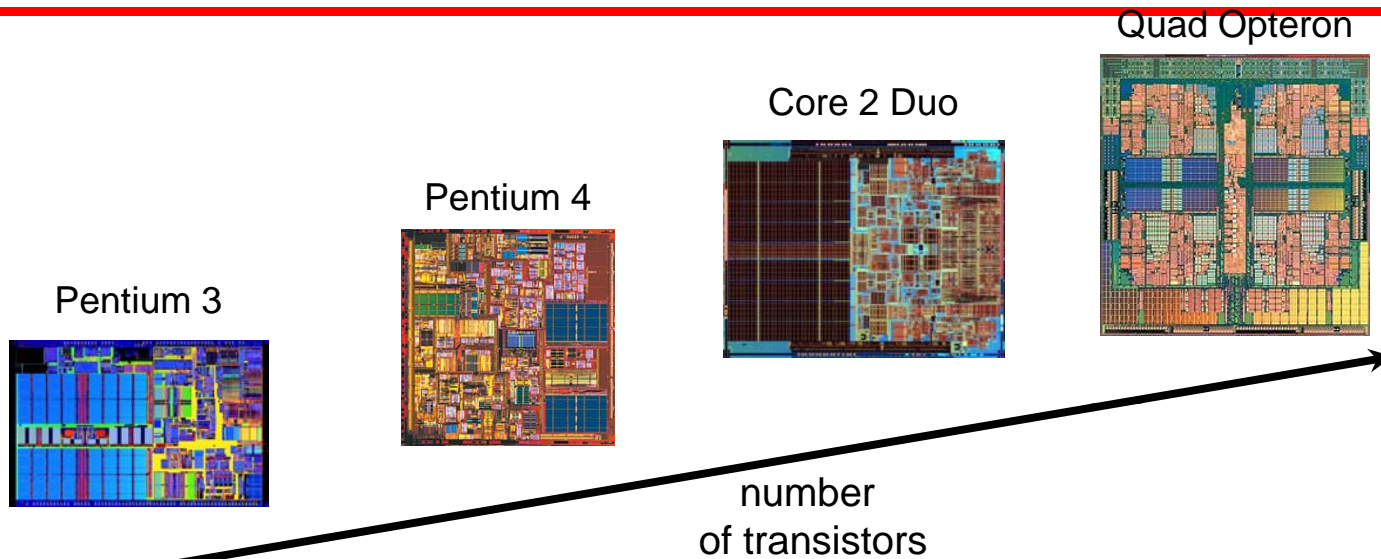
Roadmap

- Spatial variation
- Exploiting spatial variation
- Temporal variation
- Minimizing temporal variation
- The road ahead

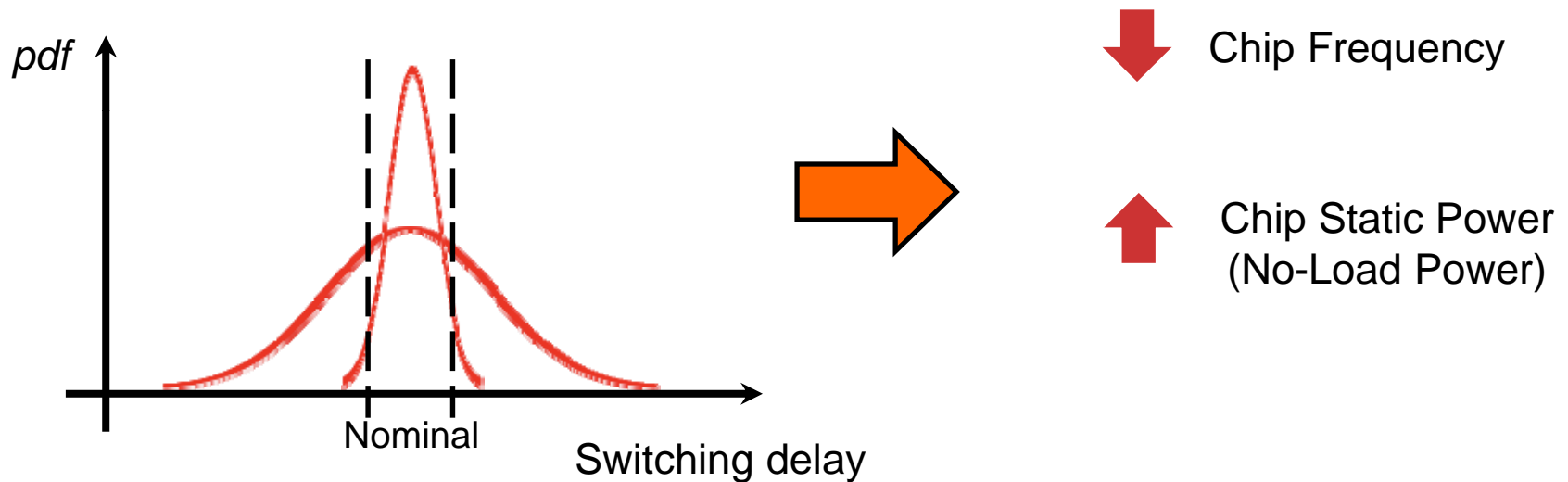
Roadmap

- Spatial variation
- Exploiting spatial variation
- Temporal variation
- Minimizing temporal variation
- The road ahead

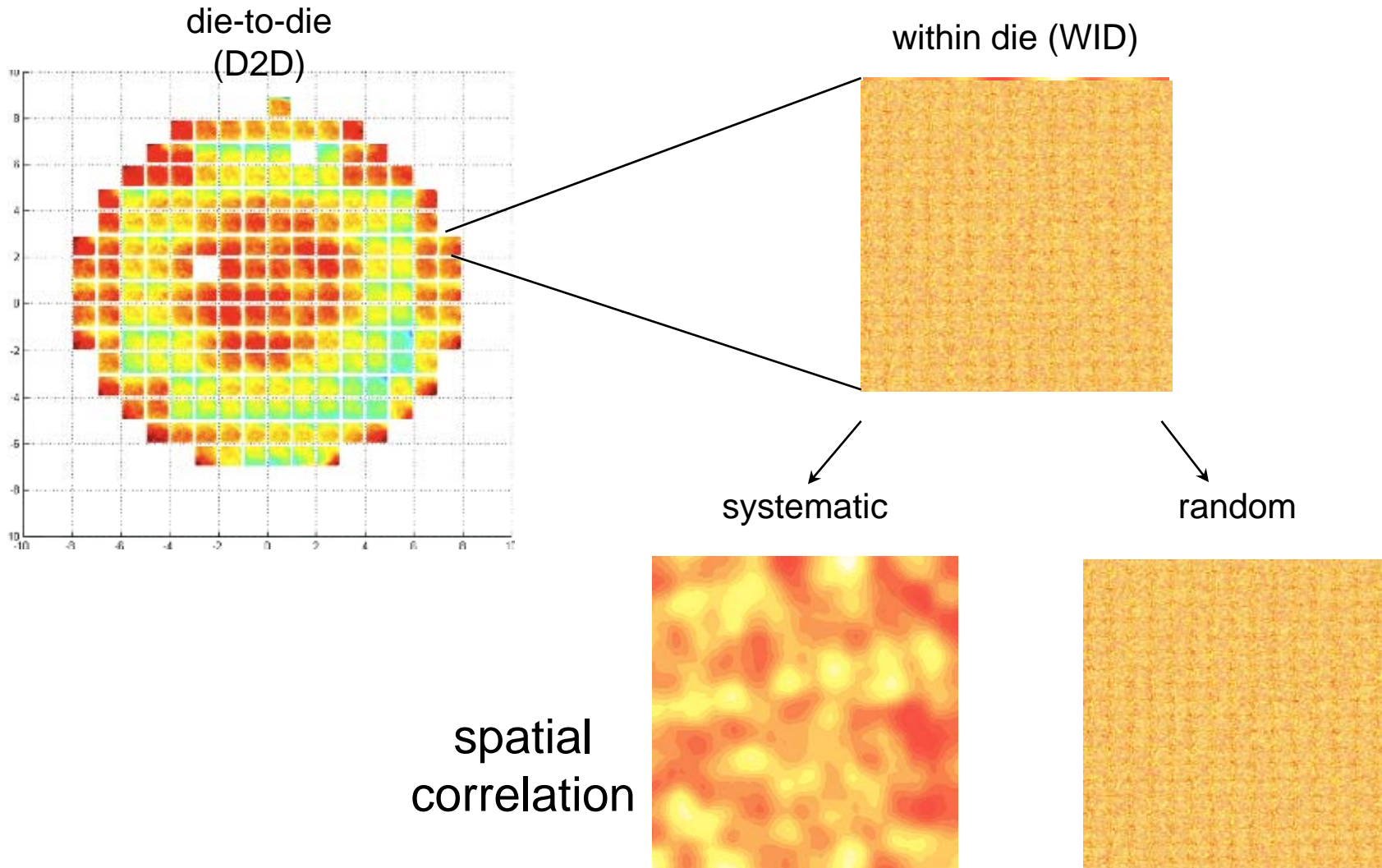
Technology Scaling Continues



Spatial Variation in Transistor Parameters



Spatial Variation Components



Result: Large Multicores Become Heterogeneous

[Teodorescu ISCA08]

- Large multicores have significant core-to-core variation
- For a 20-core multicore at 32nm (year 2010)

Static power (no-load power):

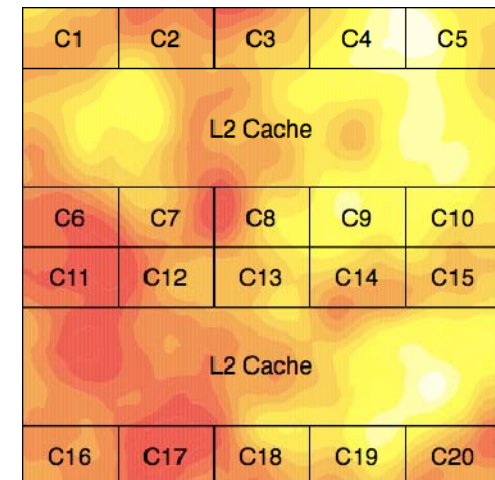
- (max/min): 1.9X avg

Total power:

- (max/min): 1.4X avg

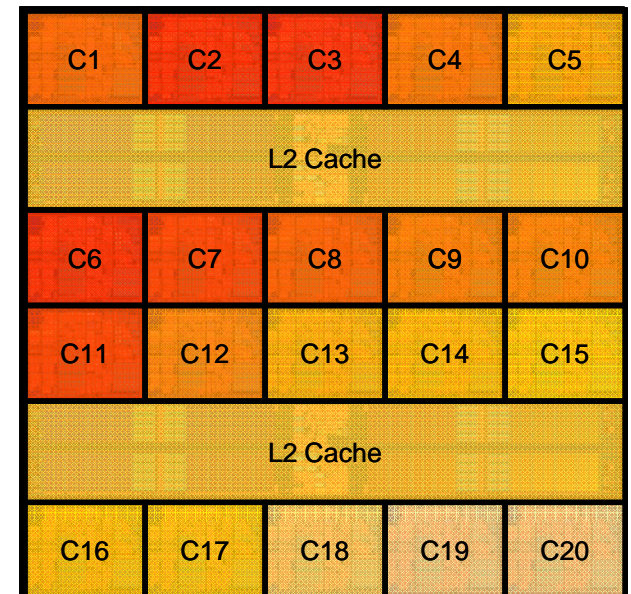
Frequency:

- (max/min) 1.3X avg



Can This Variation be Exploited?

- Traditionally: run all cores at the same frequency (slowest core)
- New multicores: Each core can run at different frequency (and voltage)
 - heterogeneous system



Roadmap

- Spatial variation
- Exploiting spatial variation
 - Variation-aware job scheduling and power management [Teodorescu ISCA08]
- Temporal variation
- Minimizing temporal variation
- The road ahead

Variation-Aware Job Scheduling

Additional information to guide scheduling decisions:

- Per core freq and static power
- Application behavior
 - Dynamic power consumption
 - Compute intensity (IPC)

Multiple possible goals

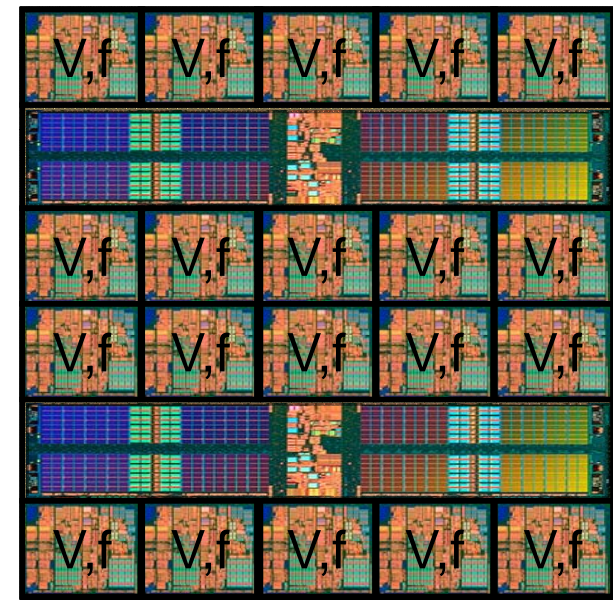


Examples of Possible Algorithms

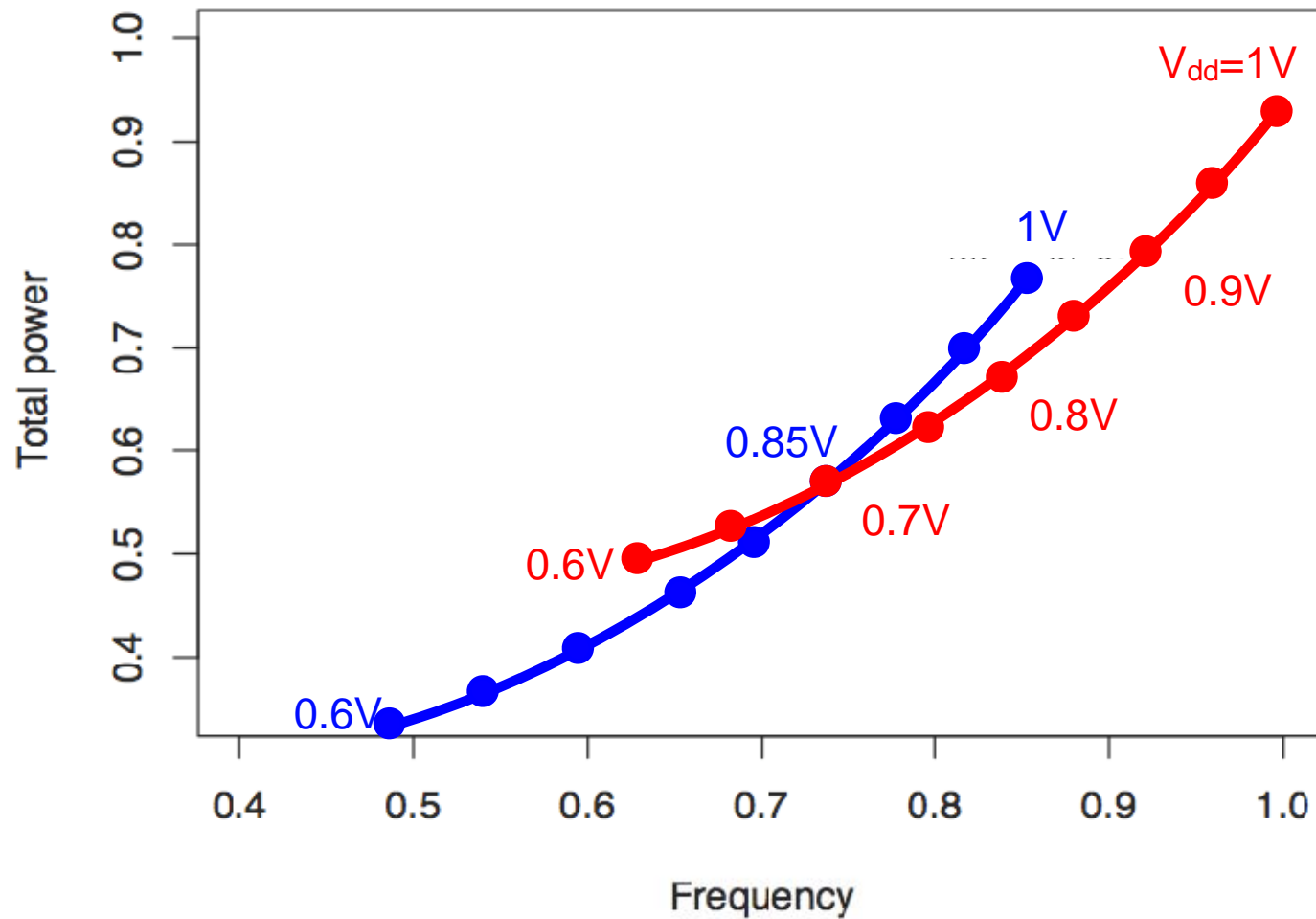
- When the goal is to reduce power consumption:
 - Assign applications with high dynamic power to low static power cores
- When the goal is to improve throughput:
 - Assign high IPC applications to high frequency cores

Variation-Aware Global Power Management

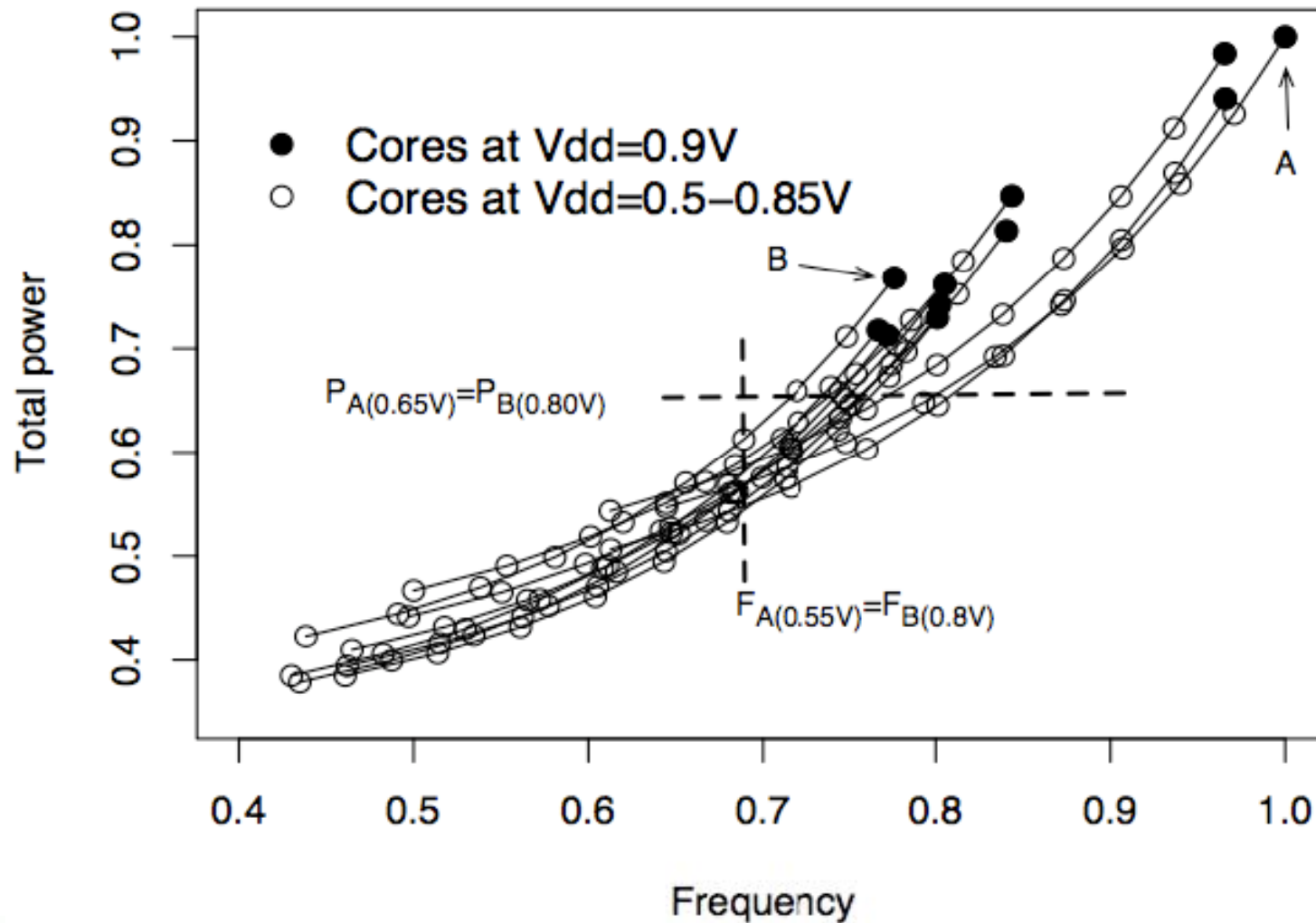
- Per core Dynamic Voltage and Frequency Scaling (DVFS)
- Challenge: find best (V,f) for each core
- Global (multicore-wide) power management solution is needed



DVFS under Variation



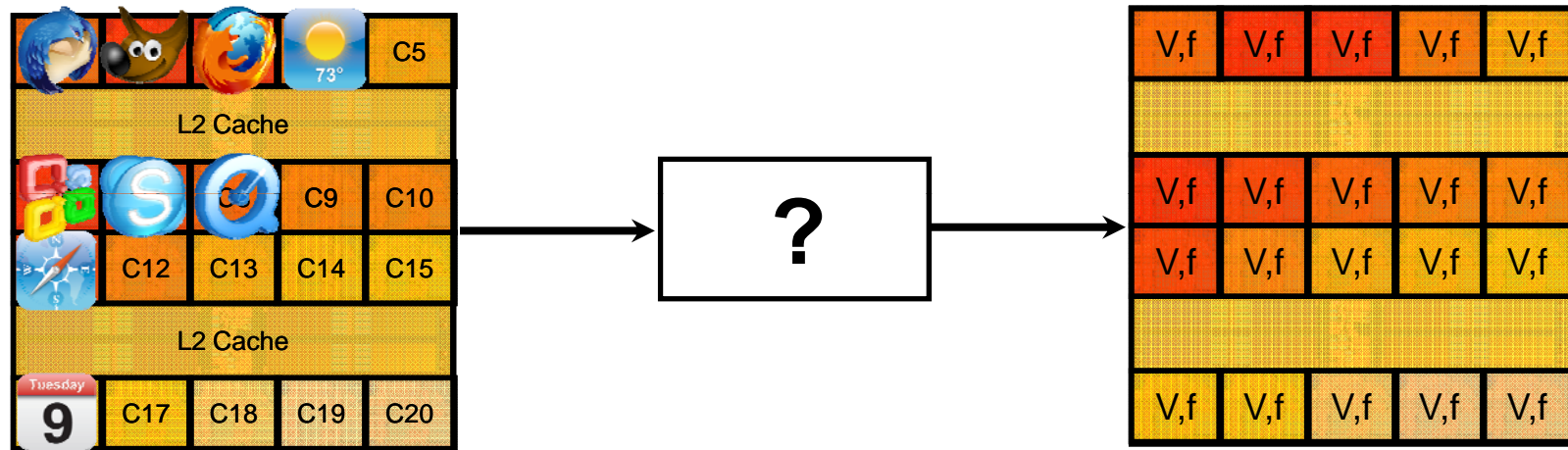
DVFS under variation (All 20 Cores)



Optimization Problem

Given a mapping of threads to cores (variation-aware):

best (V_i, f_i) of each core



- **Goal:** maximize system throughput
- **Constraint:** keep total power below budget

50W



75W



100W

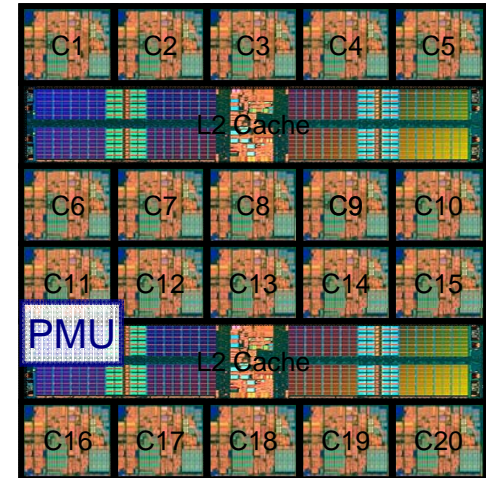


Possible Solutions

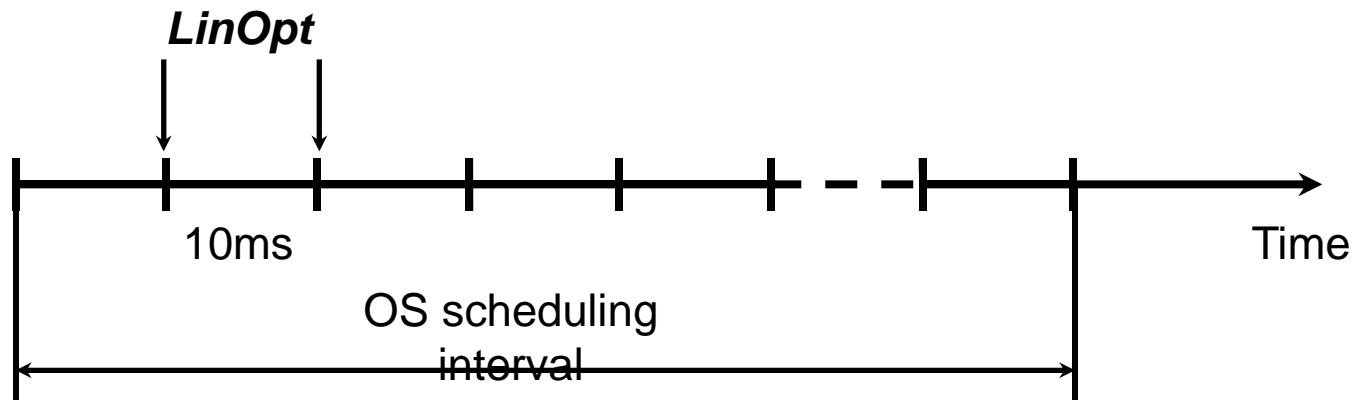
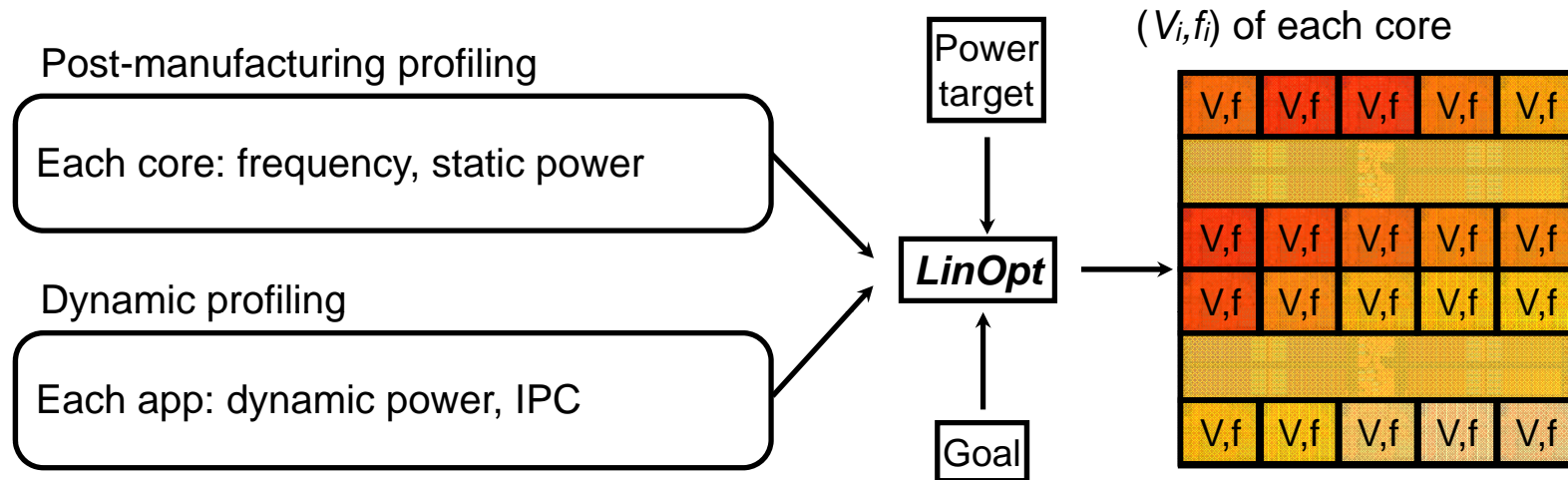
- Exhaustive search: too expensive
- Simulated annealing:
 - Not practical at runtime
- Linear programming (***LinOpt***)
 - Simpler, faster
 - Requires some approximations

LinOpt Implementation

- *LinOpt* works together with the OS scheduler
 - OS scheduler maps applications to cores
 - *LinOpt* then finds (V, f) settings for each core
- *LinOpt* runs periodically:
 - As a system process on a core, or...
 - Power management unit (PMU) - e.g., *Foxton*



LinOpt Implementation

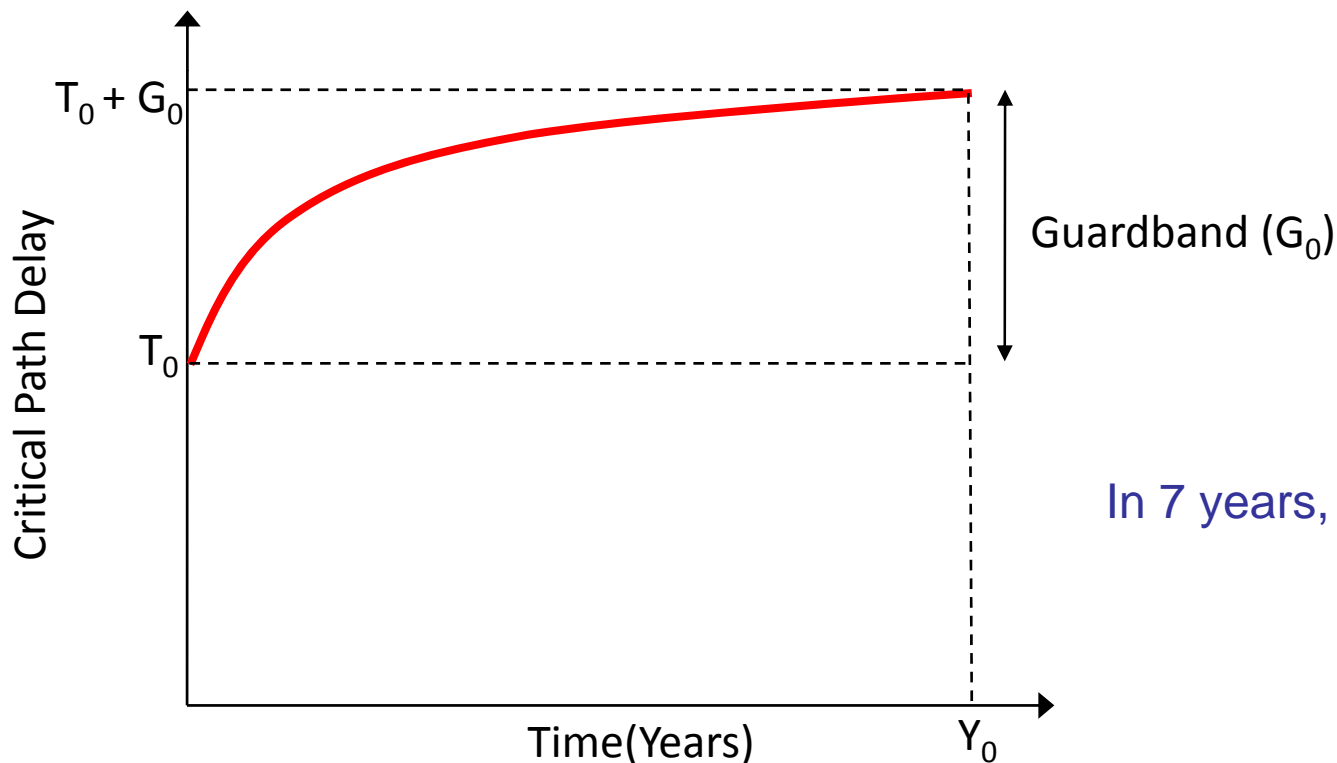


Roadmap

- Spatial variation
- Exploiting spatial variation
- Temporal variation
- Minimizing temporal variation
- The road ahead

Temporal Variation: Aging

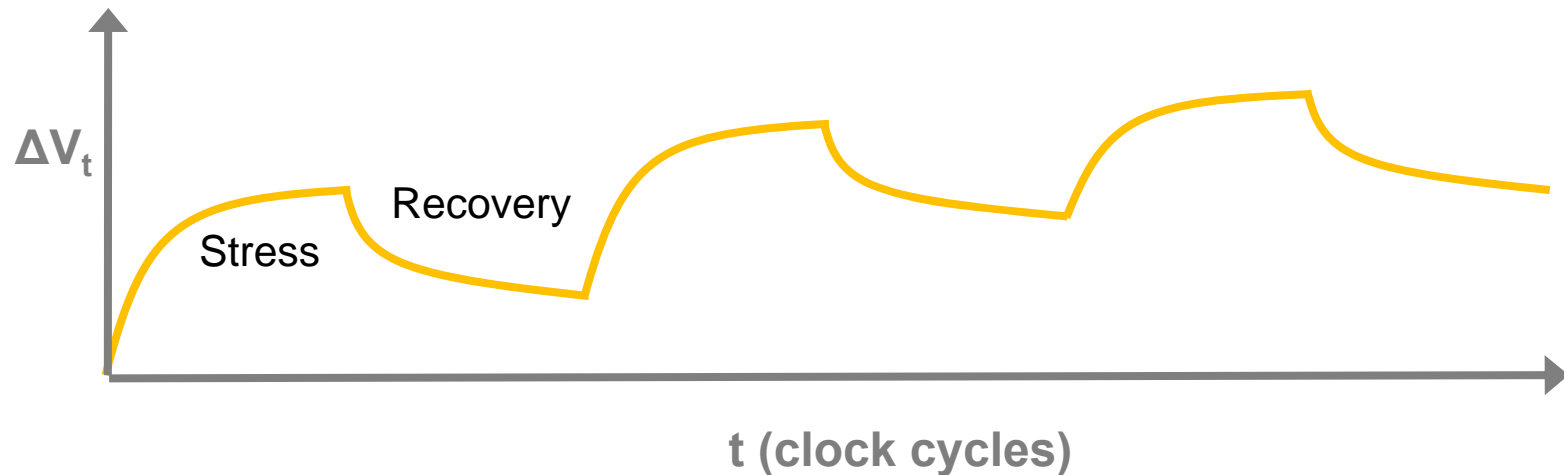
- Transistor delay gradually increases with time under normal use
- Processor critical paths take longer



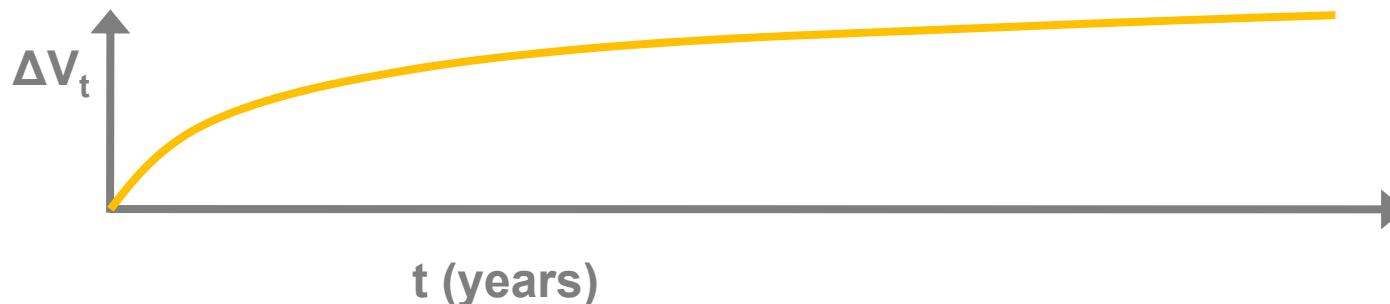
In 7 years, 5-20%??

Aging: Two Key Mechanisms

- PMOS: Negative bias temperature instability (NBTI)



- NMOS: Hot Carrier Injection (HCI)



What Affects Aging Rate?

[Tiwari MICRO 08]

Factor	NBTI	HCI
V_{dd}	exponential	exponential
T	exponential	linear
f, activity (α)	---	linear

Roadmap

- Spatial and variation
- Exploiting spatial variation
- Temporal variation
- Minimizing temporal variation
 - Aging-driven job scheduling and voltage changes
[Tiwari MICRO08]
- The road ahead

Proposed Approaches

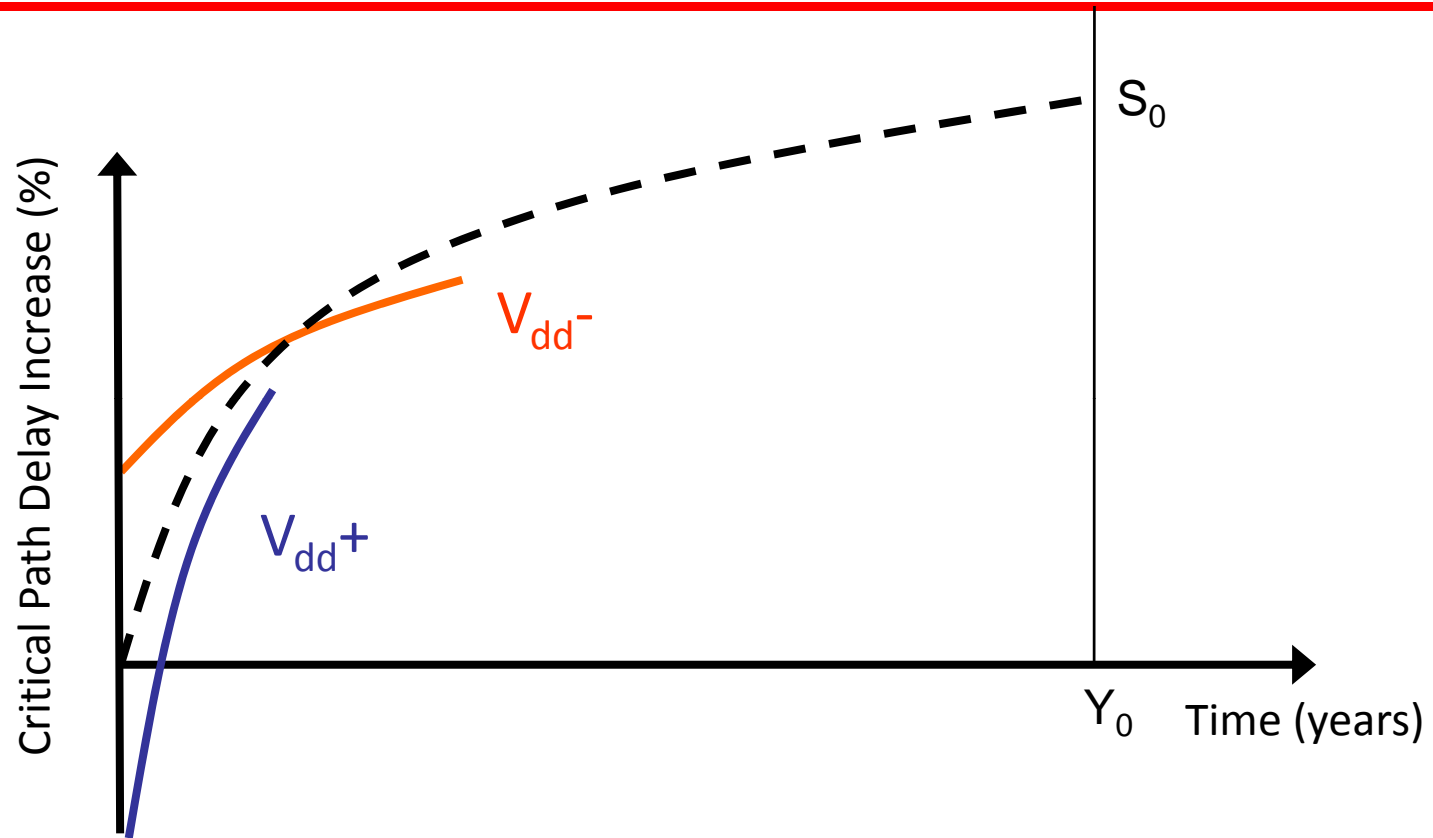
Factor	NBTI	HCI
V_{dd}	exponential	exponential
T	exponential	linear
α, f	---	linear

- Aging driven job scheduling:
 - Temperature, activity (α)
- Voltage changes at key times of processor service life
 - Changes V_{dd}
 - High impact on temperature

Aging Driven Job Scheduling

- Different policies are possible
- A possible one:
 - Send **aging-intensive** applications to the **faster** cores
 - High-T, high activity apps
 - Fast aging of cores that have more room to age
 - Send **less aging-intensive** applications to **slower** cores
 - Low-T, low activity, memory bound apps
 - Slow aging of cores that have less room to age

Voltage Changes (Vdd)



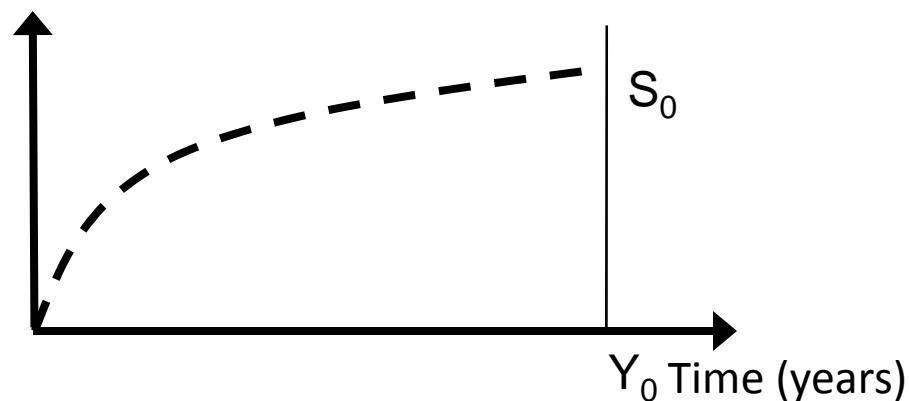
No frequency change

Big Picture

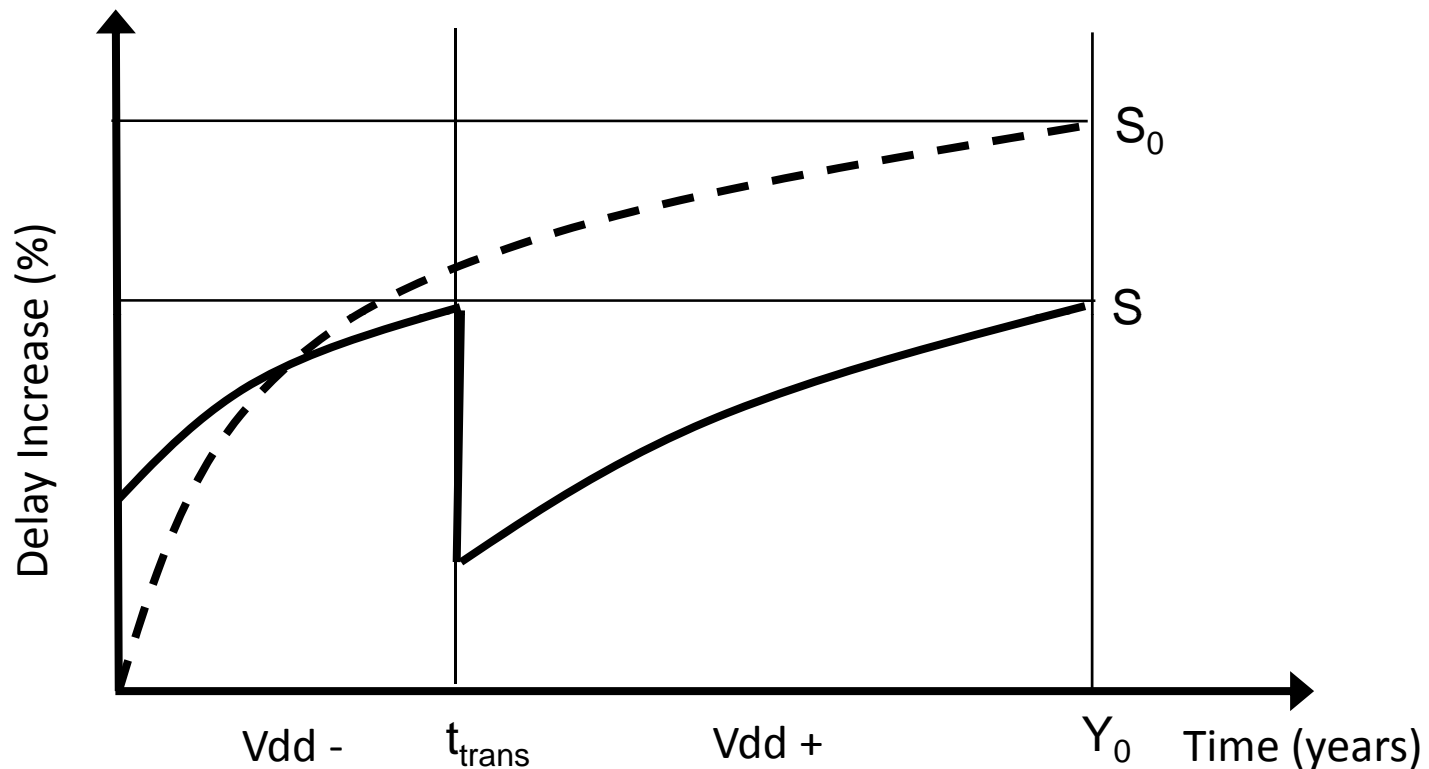
- $V_{dd} -$
 - Increases delay of logic paths
 - Slows down aging rate
- $V_{dd} +$
 - Reduces delay of logic paths
 - Increases aging rate

When to Apply These Techniques?

- Aging rate is higher in beginning and lower at end
- Apply V_{dd-} in beginning
 - Slow down aging the most
 - Have room to slow down paths
- Apply V_{dd+} at end
 - Little impact on aging rate anyway
 - Reduce logic path delays



Impact on the Aging Curve



- Need less guardband ($S\%$ rather than $S_0\%$)
- Can cycle it at higher frequency from the beginning

Roadmap

- Spatial Variation
- Exploiting spatial variation
- Temporal variation
- Minimizing temporal variation
- The road ahead

The Road Ahead (I)

- Multicores will become more **dynamic**
 - Chips will come with many sensors (power, temp, activity)
 - Multiple f domains common (AMD), multiple V domains likely
 - Control can be by a HW controller (Itanium Foxton) or in SW
- Homogeneous multicores appear as **heterogeneous**
 - Driven by single-thread performance
 - Intel's Turbo Mode
 - Perhaps forms of Core-fusion

The Road Ahead (II)

- Designs will become more **aggressive**
 - Use shorter guardbands and target the common usage
 - Rely on on-chip aging sensors to seamlessly lower performance
 - Example: Turbo Mode
- Inexpensive opportunities for **system software** to control the **heterogeneity** of homogeneous platforms
 - Variation-aware job scheduling and power management
 - Changes in V_{dd} to slow down aging

Intrinsic Heterogeneity in Multicores Due to Process Variation and Core Aging

Josep Torrellas

Department of Computer Science
University of Illinois at Urbana-Champaign
<http://iacoma.cs.uiuc.edu>

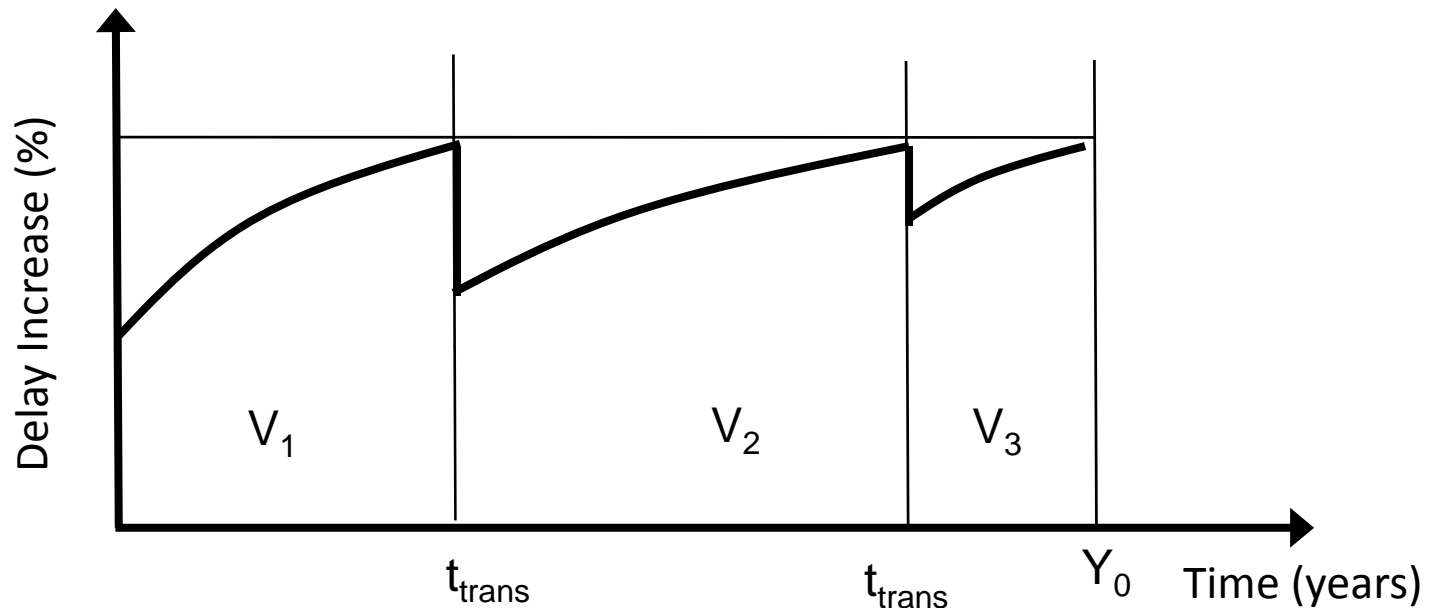
Supported by NSF, Intel and IBM



Multiprocessor Architectures for Speculative Multithreading
Josep Torrellas, University of Illinois



Can Have Multiple V Changes

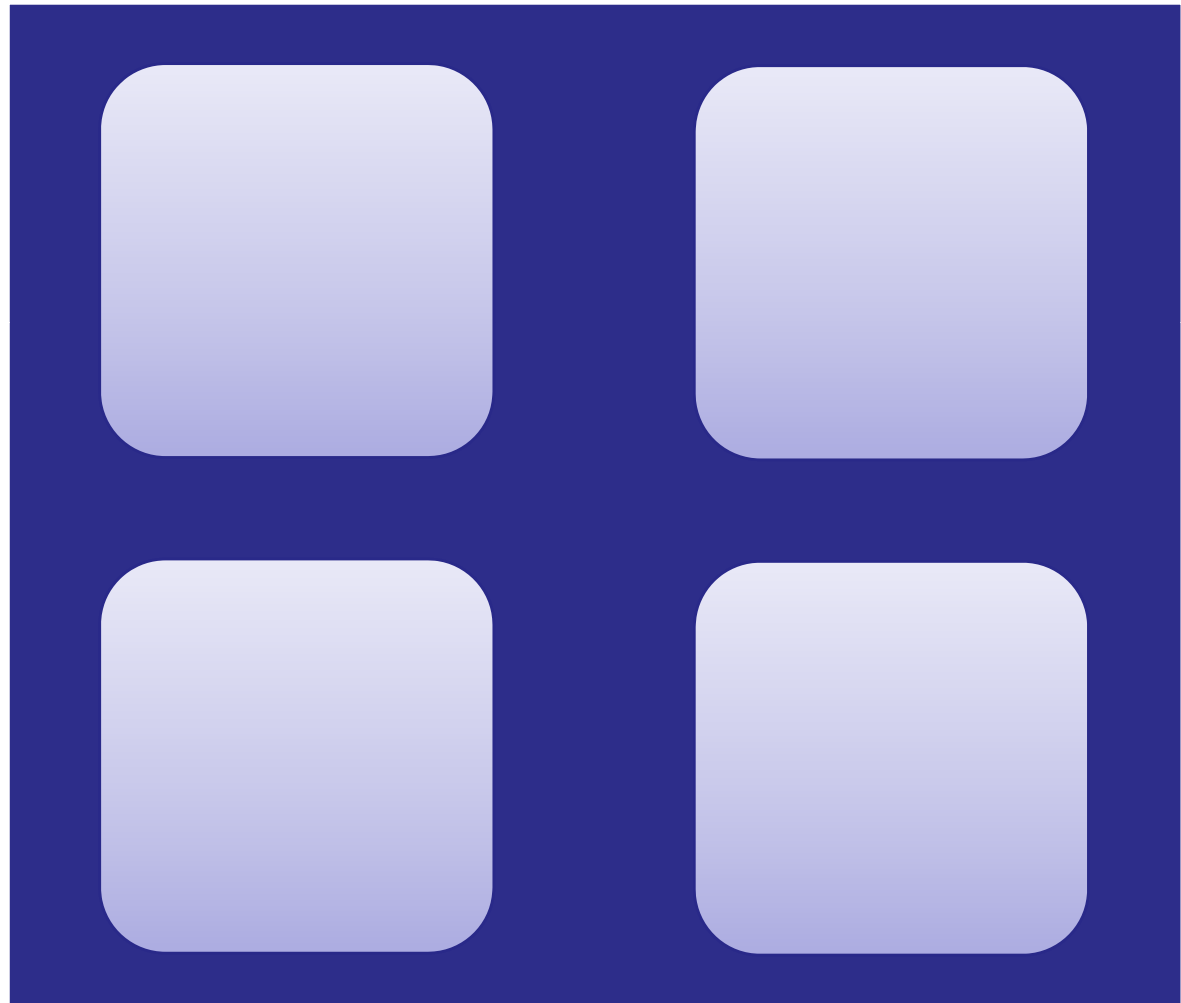


- Proposal:
 - Manufacturer programs a schedule of V increases with time based on typical usage (no f change)
 - On-chip age sensors can modify the schedule

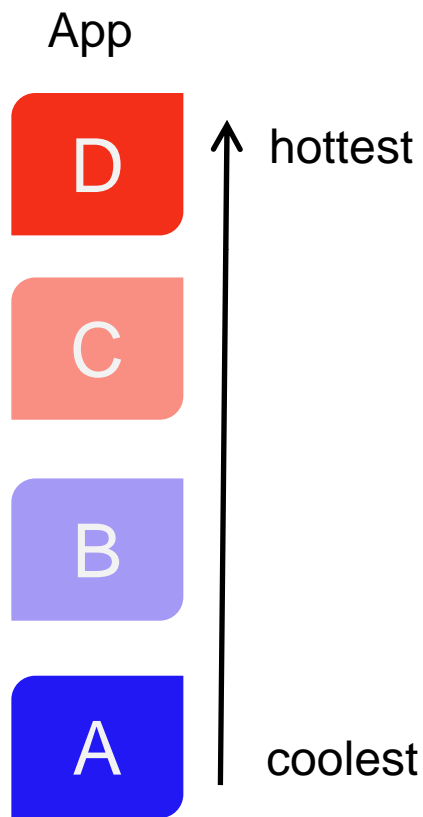
Current Results

- Variation-aware job scheduling and power management
 - Improves multicore throughput for a given power budget by 12-17%
- Slowing down aging
 - Improves frequency of multicore by 12%
 - Regains > 50% of performance losses due to aging
- Low hardware overhead
 - Change V, job scheduling

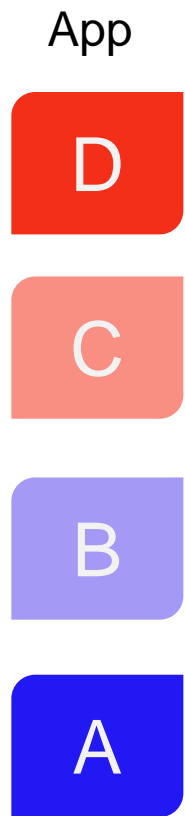
Example



Example



Example



What Affects Aging Rate?

[Tiwari MICRO 08]

Factor	NBTI	HCI
$V_{dd} - V_t$	exponential	exponential
T	exponential	linear
f, activity (α)	---	linear

- Change aging rate with
 - Adaptive Supply Voltage (ASV):
 - Changes V_{dd}
 - Exponential impact on T
 - Adaptive Body Bias (ABB):
 - Changes V_t
 - Exponential impact on T