# Extreme Scale Computer Architecture: Energy Efficiency from the Ground Up

**Josep Torrellas**

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

HARSH Workshop, February 2013

# The State of The Art

**Blue Waters**
~1 PF sustained
>300,000 cores
>1 PB of memory
>10 PB of disk storage
~500 PB of archival storage
>100 Gbps connectivity

10-20 MW

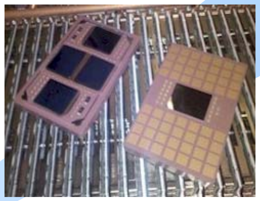
**Blue Waters Building Block**
32 IH server nodes
32 TB memory
256 TF (peak)
4 Storage systems
10 Tape drive connections

**IH Server Node**
8 MCM's (256 cores)
1 TB memory
8 TF (peak)

*Fully water cooled*

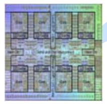
10MW = $10M per year electricity

800 W



**Multi-chip Module**
4 Power7 chips
128 GB memory
512 GB/s memory bandwidth
1 TF (peak)

**Router**
1,128 GB/s bandwidth



**Power7 Chip**
8 cores, 32 threads
L1, L2, L3 cache (32 MB)
Up to 256 GF (peak)
*45 nm technology*

## Wanted: Energy-Efficient Computing

- Extreme Scale computing: 100-1000x more capable for the same power consumption and physical footprint
  - Exascale ($10^{18}$ ops/cycle) datacenter: 20MW
  - Petascale ($10^{15}$ ops/cycle) departmental server: 20KW
  - Terascale ($10^{12}$ ops/cycle) portable device: 20W

## Energy-Efficiency Gap

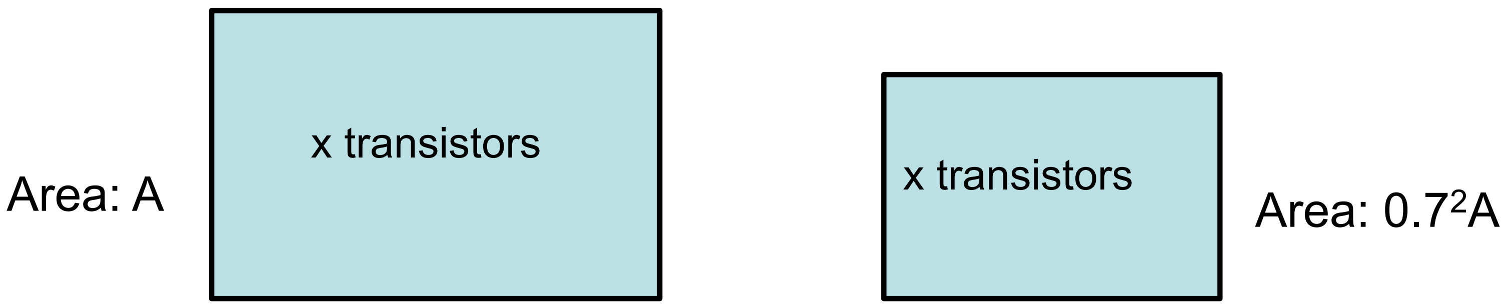
- Goal:
  - 20W Tera-Op (sustained)
  - 20 pJoules/operation

- In comparison:
  - IBM Power7 released 2010:  MCM 800W for 1TFlop Peak
    - Problem is harder than it looks:
      - Machines spend much of the energy transferring data
      - Minimizing E in data transfer, not ALU op is the challenge

# Recap: How Did We Get Here?

- Ideal Scaling (or Dennard Scaling): Every semicond. generation:
  - Dimension: 0.7
  - Area of transistor: 0.7x0.7 = 0.49
  - Supply Voltage ($V_{dd}$), C: 0.7
  - Frequency: 1/0.7 = 1.4

$$P_{dyn} \propto CV_{dd}^2 f$$



Power density: $CV_{dd}^2 f/A$

Power density: $0.7C\ 0.7^2V_{dd}^2\ 1.4f/0.7^2A$
$= CV_{dd}^2 f/A$

**Constant power density**

## Recap: How Did We Get Here ? (II)

- Real Scaling: $V_{dd}$ does not decrease much.
  - If too close to threshold voltage ($V_{th}$) → slow transistor
  - Delay of transistor is inversely prop to ($V_{dd}$ - $V_{th}$)

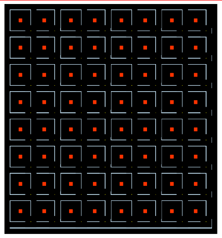
$$T_g \propto \frac{V_{dd} L_{eff}}{\mu (V_{dd} - V_t)^{\alpha}}$$

  - Dynamic power density increases with smaller tech

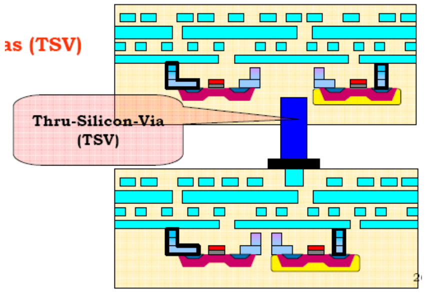
- Additionally: There is the static power

**Power density increases rapidly**

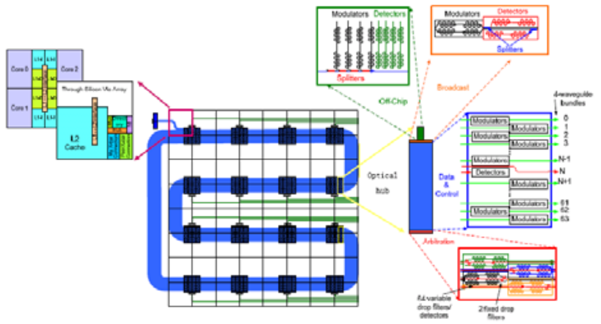
## Design for E Efficiency from the Ground Up



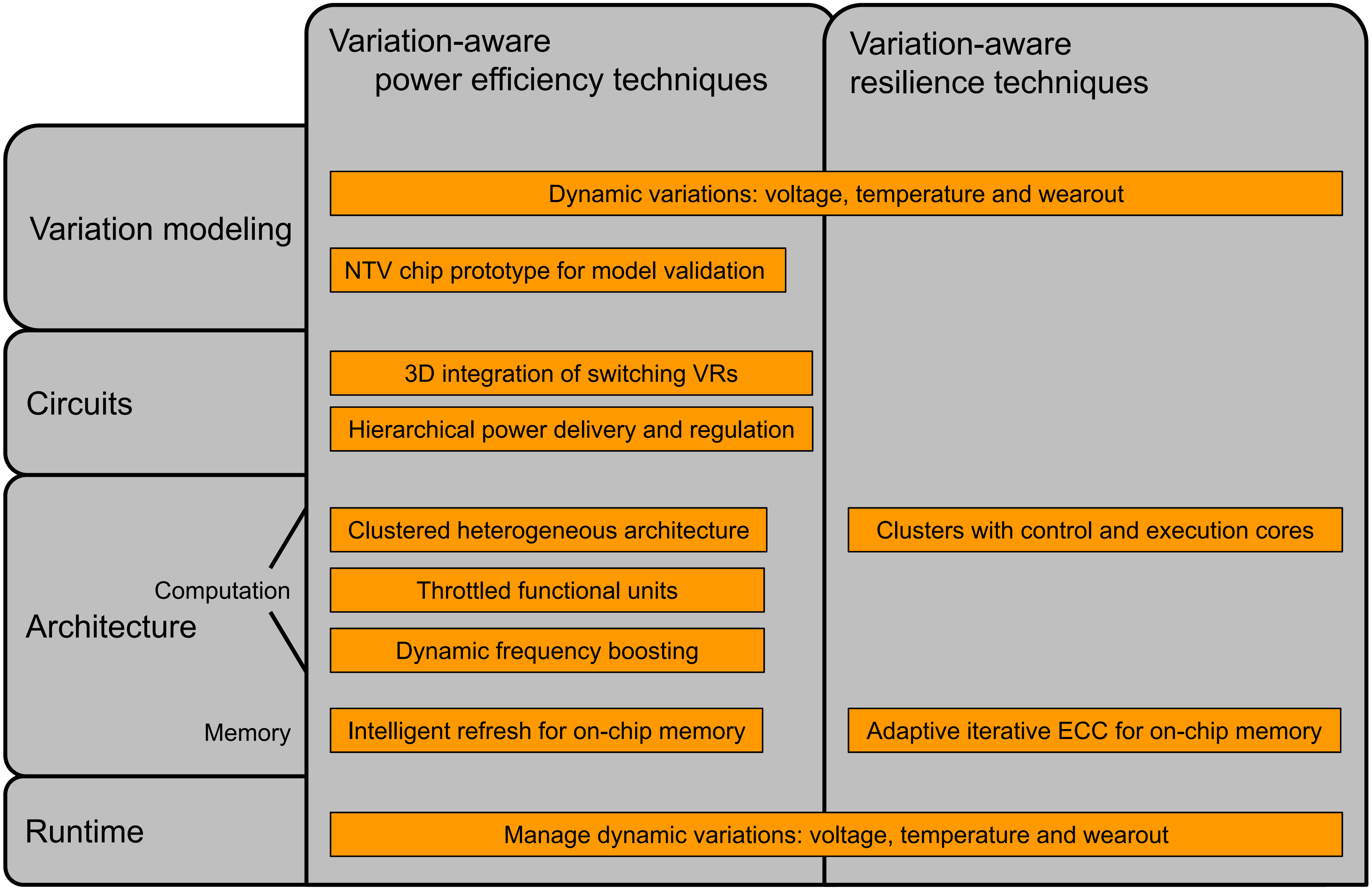
- New designs for manycore chips:
  - Efficient support for high concurrency
  - Data transfer minimization
  - Many techniques for energy efficiency can affect resilience



- New technologies:
  - Low supply voltage ($V_{dd}$) operation
  - Efficient on-chip voltage regulation
  - 3D die stacking
  - Resistive memory
  - Photonic interconnects



# A View of the Work [Kim, Teodorescu, Torrellas, Karpuzcu]


Variation-aware power efficiency techniques
Variation-aware resilience techniques
Variation modeling
Dynamic variations: voltage, temperature and wearout
NTV chip prototype for model validation
Circuits
3D integration of switching VRs
Hierarchical power delivery and regulation
Architecture
Computation
Clustered heterogeneous architecture
Clusters with control and execution cores
Throttled functional units
Dynamic frequency boosting
Memory
Intelligent refresh for on-chip memory
Adaptive iterative ECC for on-chip memory
Runtime
Manage dynamic variations: voltage, temperature and wearout

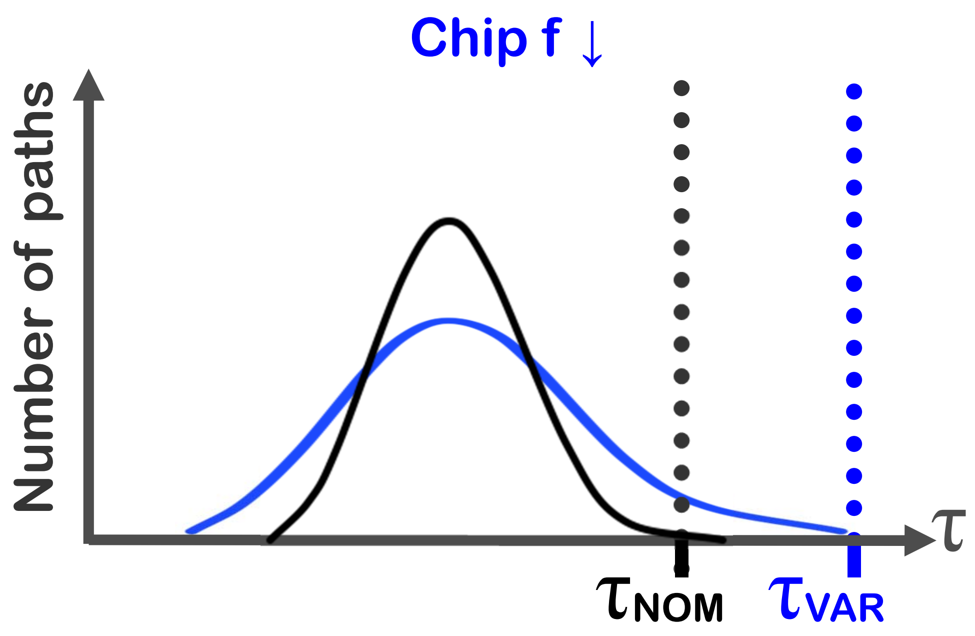
## Low Voltage Operation

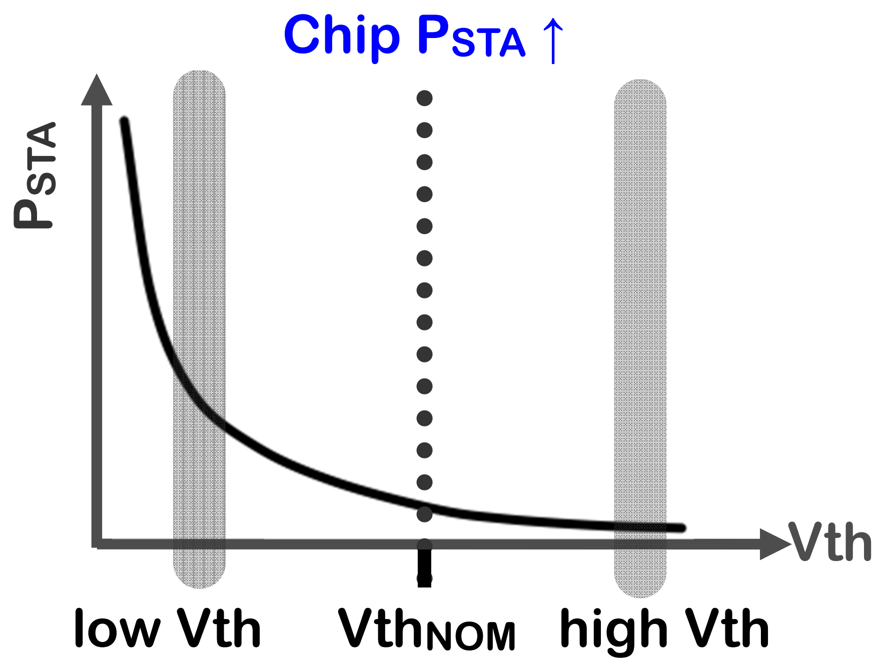
- $V_{dd}$ reduction is the best lever for energy efficiency:
  - Big reduction in dynamic power; also reduction in static power
- Reduce $V_{dd}$ to bit higher than $V_{th}$ (Near Threshold Voltage--NTV)
  - Corresponds to $V_{dd}$ of about 0.55V rather than current 1V

- Advantages:
  - Potentially reduces power consumption by more than 40x
- Drawbacks:
  - Lower speed (1/10)
  - Increase in gate delay variation

## Basics of Parameter Variation

- Deviation of device parameters from nominal values: eg Vth, Leff

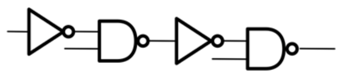


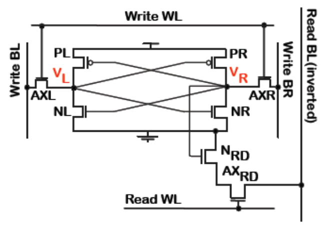


Additionally: Same ΔVth causes higher Δf and ΔP at NTV

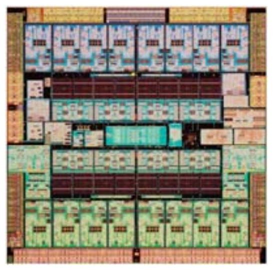
# VARIUS-NTV Model [DSN-2012]

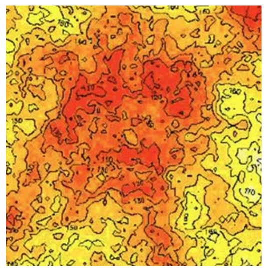
- Models variation in frequency and power at a range of Vdd
- Applies to logic and on-chip memory
- Suitable for architects



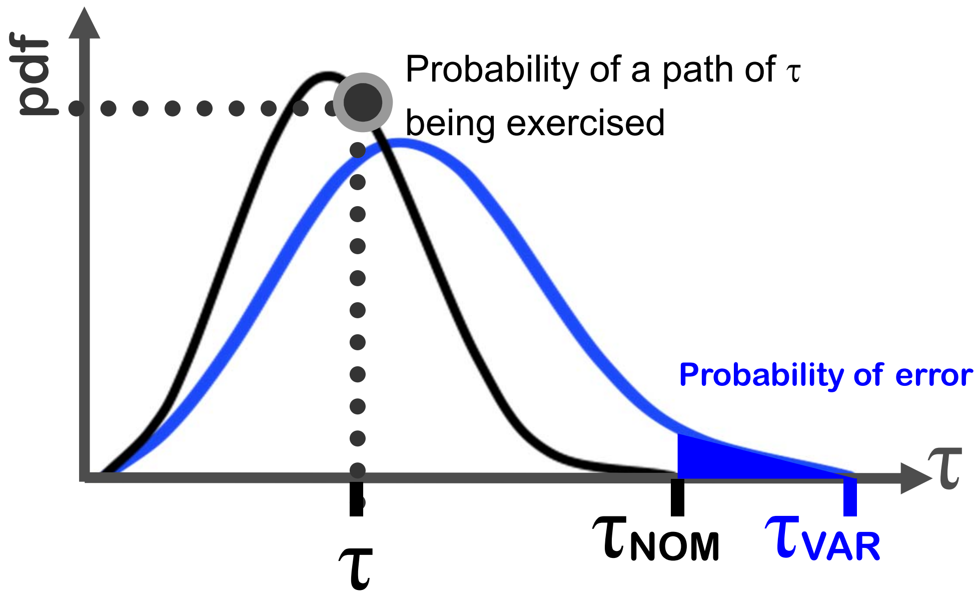


Systematic + random variation

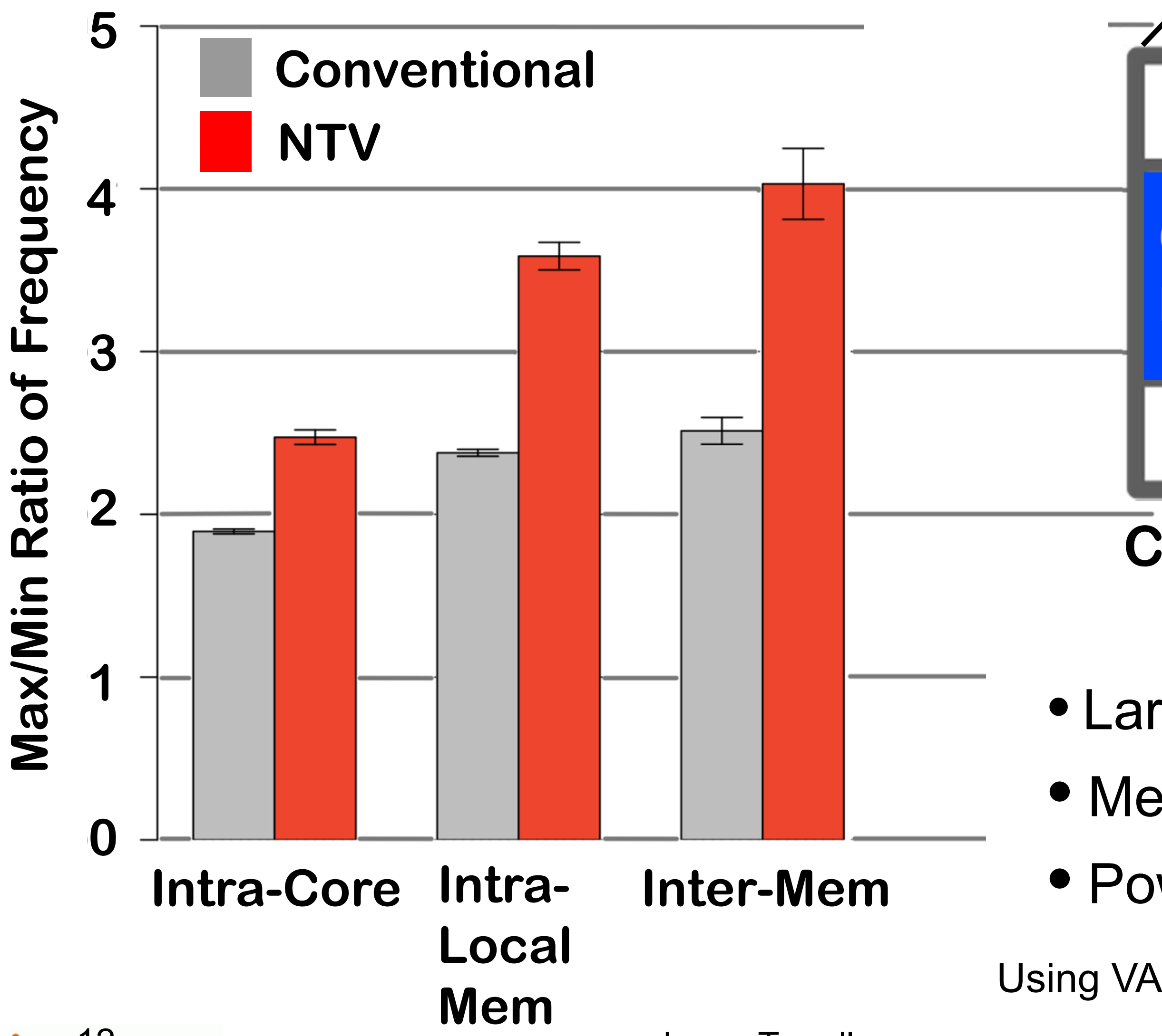


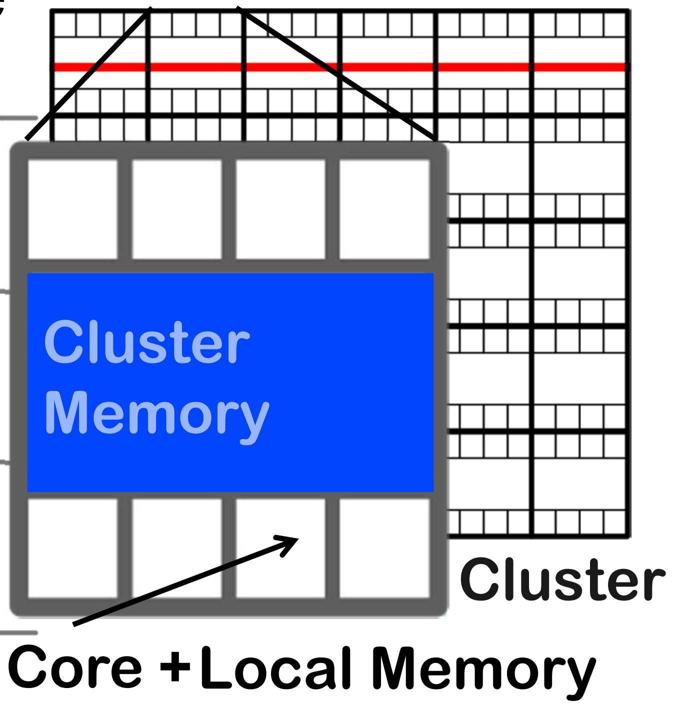


- Also gives the (timing) error rate at a given f, Vdd



# Variation in Thrifty Manycore



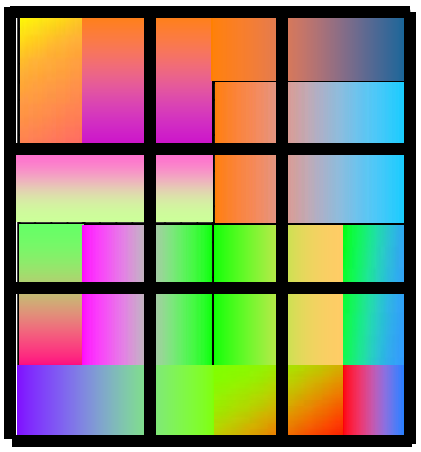


- Larger f variation at NTV
- Memories more vulnerable
- Power varies as well

Using VARIUS-NTV by Karpuzcu et al

## Multiple $V_{dd}$ Domains at NTV: Hardly Effective

- On chip regulators have a high power loss (10+%)
- To reduce costs, only coarse-grain (multiple-core) domains
  - Already has variation inside the domain
- Small $V_{dd}$ domain more susceptible to load variations
  - Larger $V_{dd}$ droops → need increase $V_{dd}$ guardband

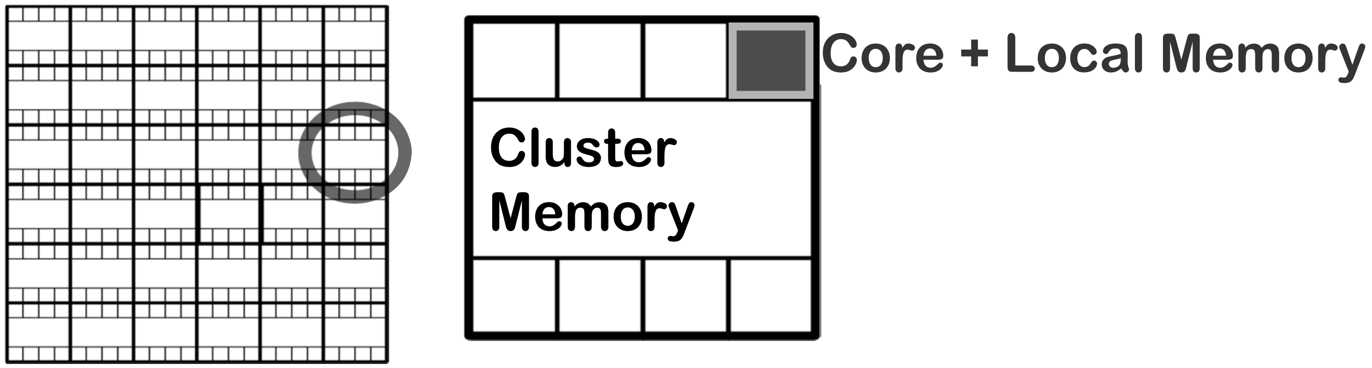


Work with:
Ulya Karpuzcu (U Minn) and
Nam Sung Kim (U Wisc)

## Propose: Energy Efficiency with a Single $V_{dd}$ Domain

One $V_{dd}$ domain, many f domains

- Simple hardware, simple & effective core allocation



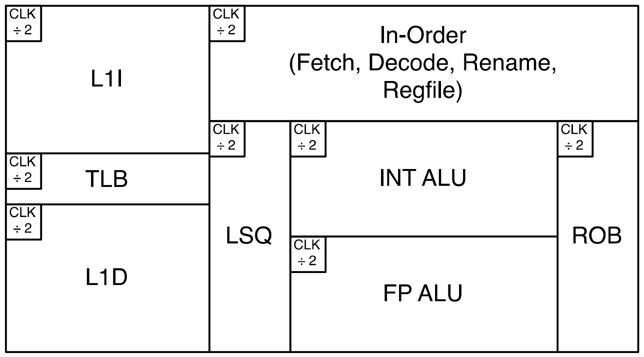
- Each cluster in the chip is a f domain
- Allocation in units of multiples of clusters called Ensembles
  - Whole ensemble clocked at a single f
- Simpler variation-aware core allocation

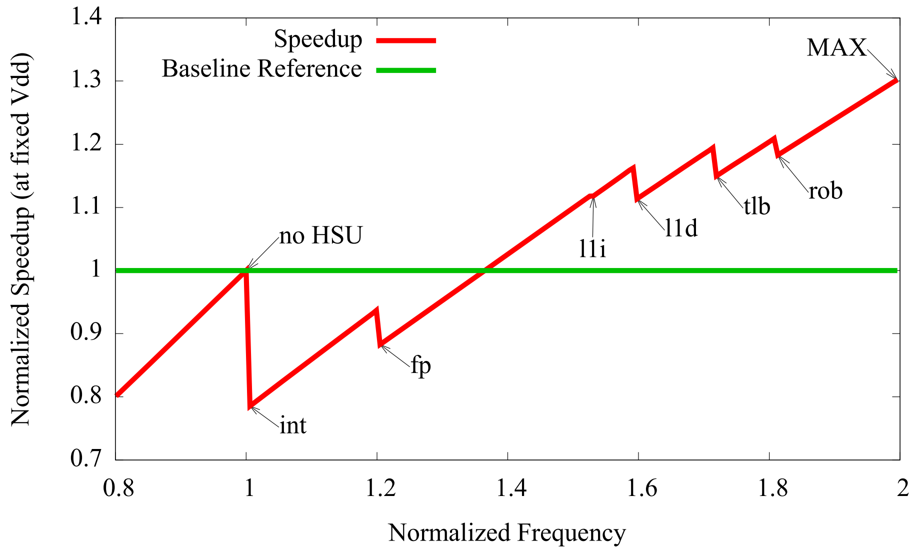
## Streamlined 1K-core Architecture

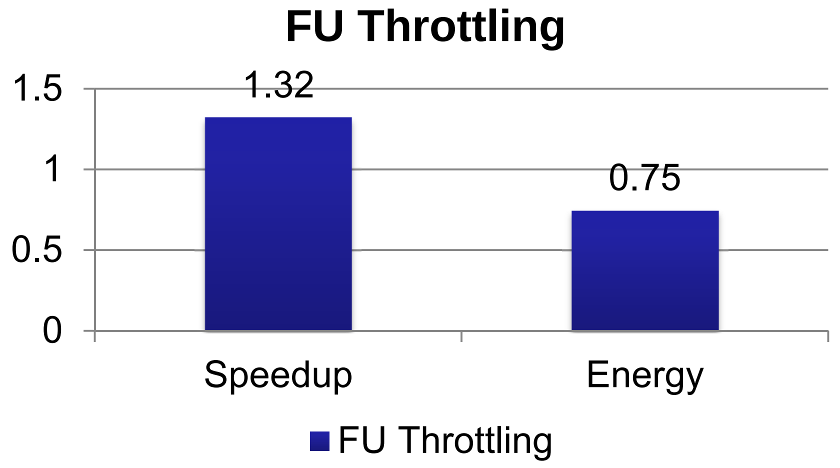
- Very simple cores (no structures for speculative execution)
- Cores organized in clusters with memory to exploit locality
- Each cluster is heterogeneous (has one large core)
- Special instructions for certain ops: fine-grain synch
- Single address space without hardware cache coherence

# Functional Unit Throttling [Miller HPCA-2012]

- Improve core frequency by throttling slow functional units
  - Functional units can run at two speeds: full and half-speed
  - Slow functional units run at half clock speed allowing core frequency to be raised



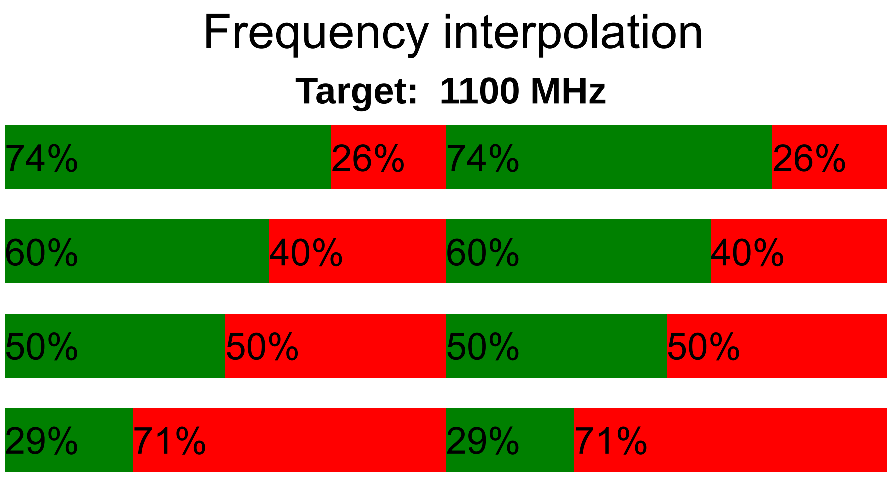


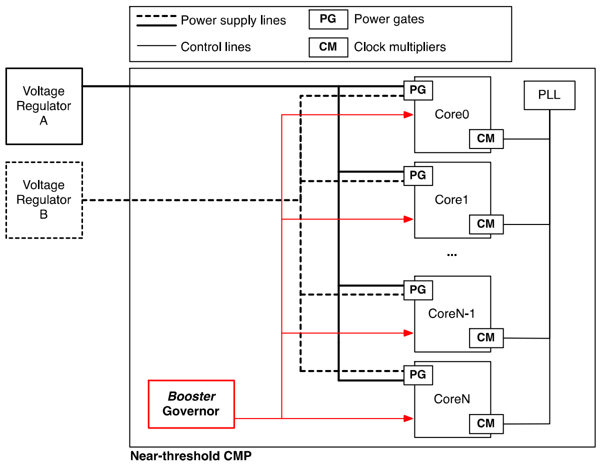


# Boosting with Dual-Vdd Rails [Miller CAL-2012]

- Reduce/eliminate frequency heterogeneity
  - Dual-Vdd set at two different low Vdds; cores switch between them
  - Slow cores most of the time on high Vdd – boosts frequency
  - Fast cores most of the time on low Vdd – saves power

| Core | Low Freq. | High Freq |
|---|---|---|
| $C_{12}$ | 775 | 2025 |
| $C_{19}$ | 650 | 1775 |
| $C_{27}$ | 575 | 1625 |
| $C_{50}$ | 425 | 1375 |



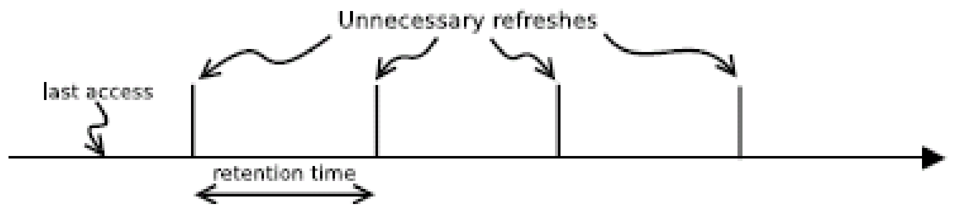


## Managing the Power of On-Chip Memories

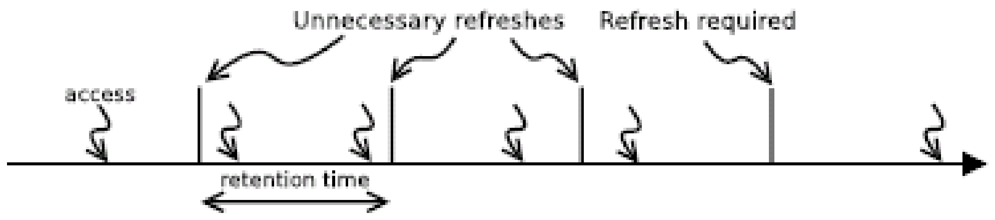
- On-chip memory leakage: major contributor of the NTV chip power
- Coarse-grained proposals are insufficient
  - Turn off some memory modules / disable cache ways / …
- Needed: power-on only the lines that contain useful data
- Proposal
  - Use on-chip memory technology that does not leak (eDRAM) --- but needs to be refreshed
  - Use fine-grain, intelligent refresh of the on-chip memory
- Great opportunity of major power savings
  - Much of the on-chip memory contains useless data!

# When Useless Refresh Happens

- Cold lines: Lines not used or used far apart in time

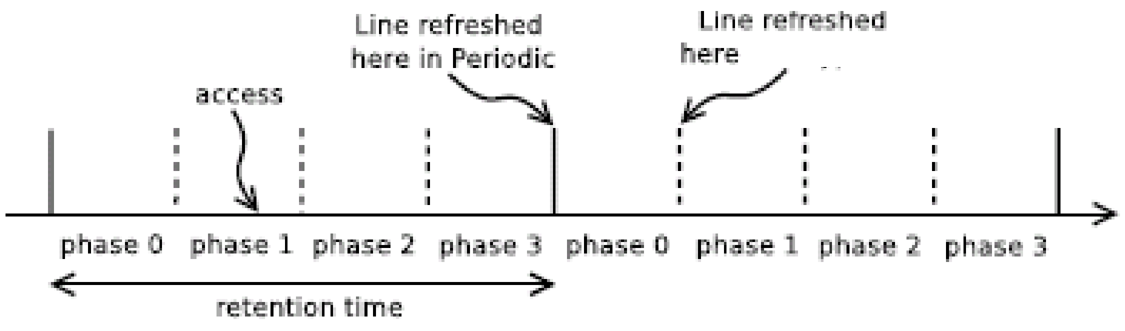


- Hot lines: Lines actively used



# Refrint: Intelligent Refresh

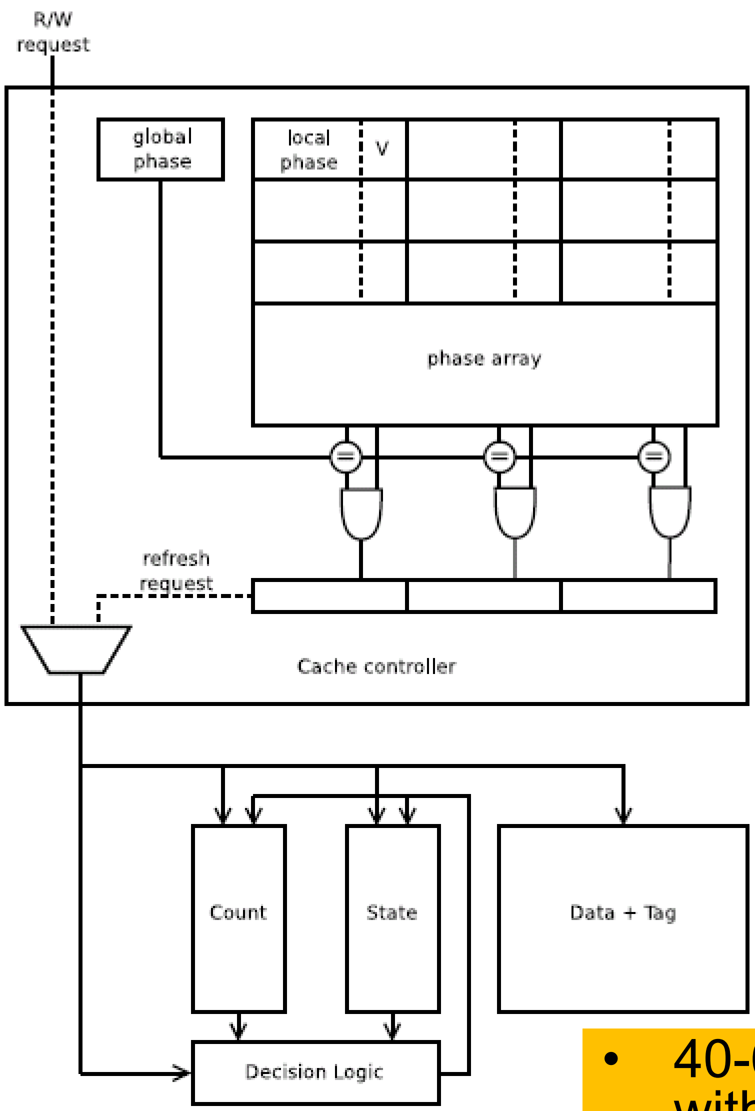
- When to refresh:
  - Divide the retention period into equal intervals called Phases
  - Maintain for each line: phase in which it was last accessed (or refreshed)
  - A line is refreshed only when the same phase arrives in the next retention period.



## Refrint: Intelligent Refresh

- What to refresh:
  - Use state of the line:
    - Valid data but give a “grace period”: WB (n,m)
      - Dirty lines refreshed n times before writeback
      - Clean lines refreshed m times before inval

## Simple Hardware



When to refresh:

- Cache controller keeps, for each line, the phase it was last refreshed/accessed
- At the beginning of phase: controller checks for lines with matching phase
- For each line: 2 bits for phase, 1 for valid

What to refresh:

- Keep a per-line countdown of refreshes
  - Reset at access
  - Decrement at refresh.
- When counter reaches zero, wb/inval

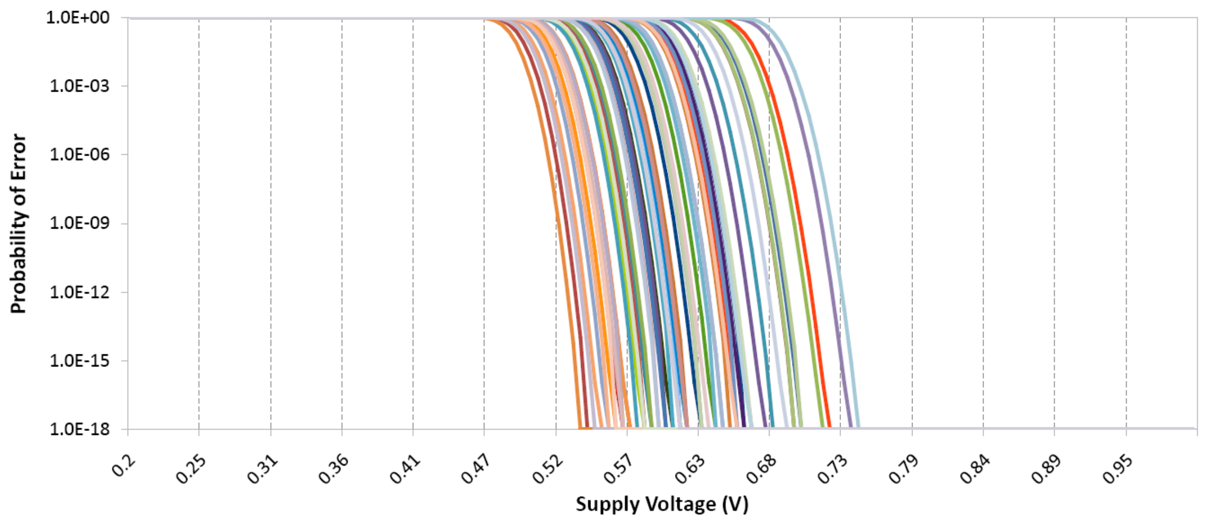
- 40-60% reduction in on-chip memory energy with no slowdown

## Network Reliability

- Networks are especially vulnerable to  variation:
  - They connect distant parts of the chip (different speed & power)
- Aggressive power savings:
  - Dynamically reduce Vdd of each router to the minimum while watching for errors
- Highly energy efficient
  - Remove Vdd margins added for variation and wearout
  - Inexpensive error detection: end-to-end

## Error Rate as Function of Vdd

- Process variation has a major impact on the routers



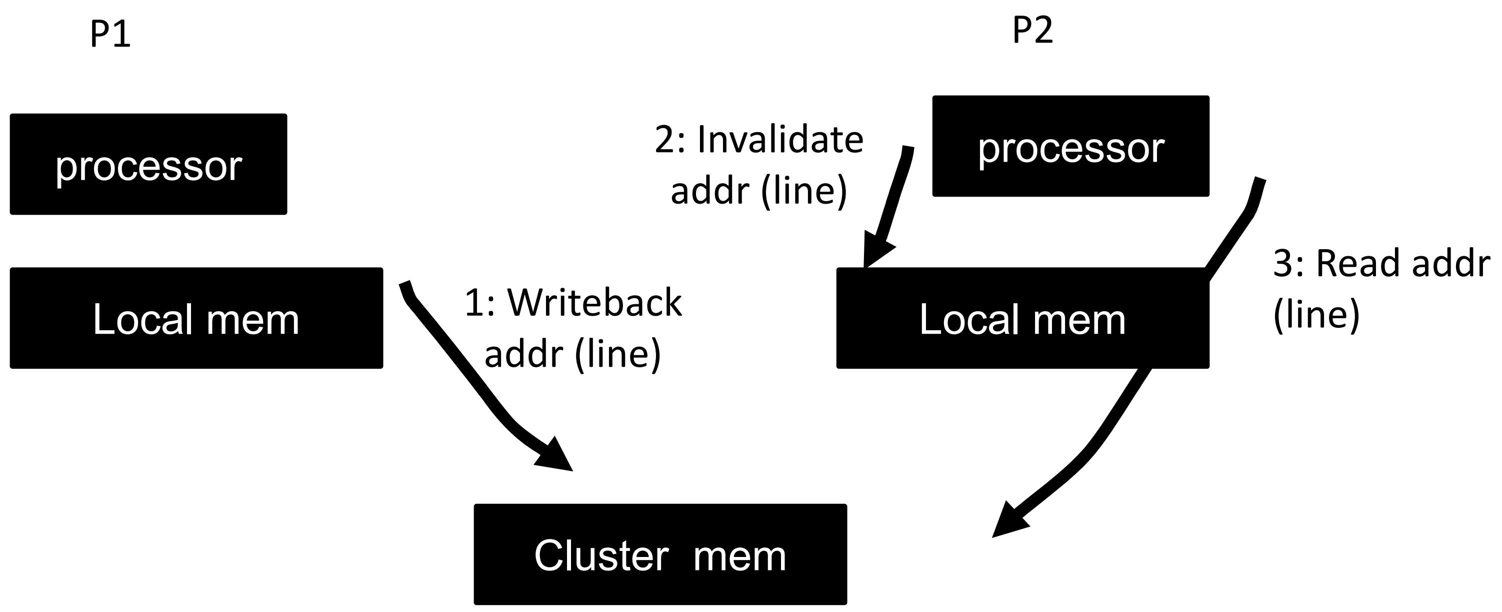
- Energy savings of 20-30% of network while keeping the system reliable
- Only 1-1.5% performance impact

## Minimizing Data Movement

- Several techniques to minimize data movement:
  - Many-core chip organization based on clusters
  - Mechanisms to manage the cache hierarchy in software
  - Simple compute engines in the mem controllers → Processing in Memory (PIM)
  - Efficient synchronization mechanisms

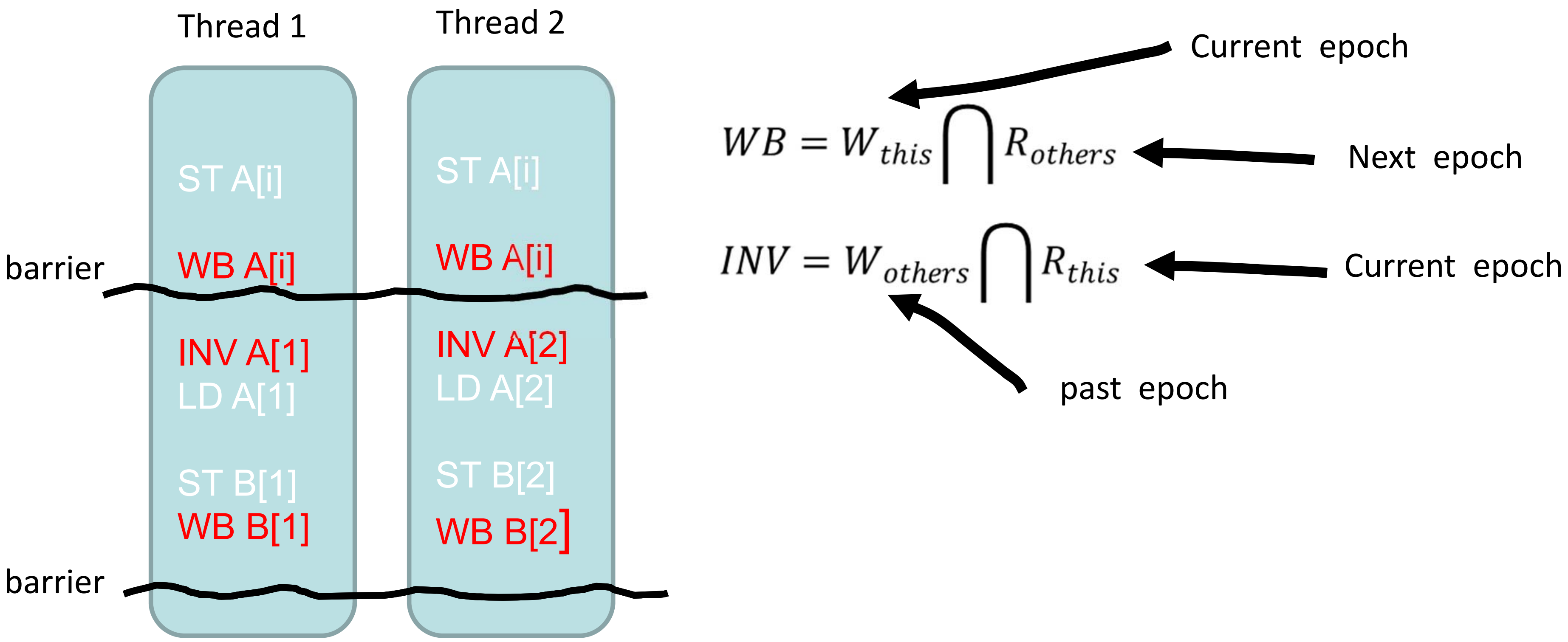
## Software Managed Caches (SMC)

- When core references data, HW brings a copy of line to cache from first level of cache it finds it in
  - May not be latest version
- Writes do not invalidate/update other copies of the line
- Need instructions to perform explicit write-back and invalidate



# SMC Programming

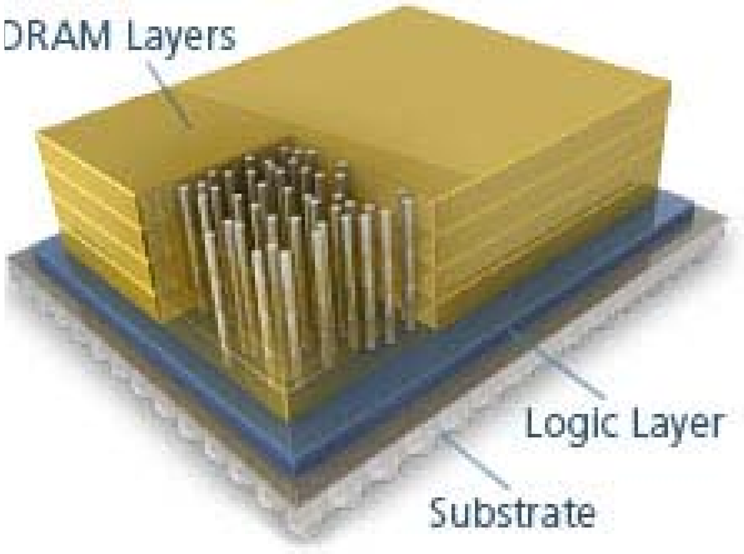
- Programmer/compiler inserts data-movement instructions at synchronization points
- Hopefully minimizes data transferred



$$WB = W_{this} \bigcap R_{others}$$

$$INV = W_{others} \bigcap R_{this}$$

## Processing in Memory



Micron's Hybrid Memory Cube (HMC) [Micron10]:

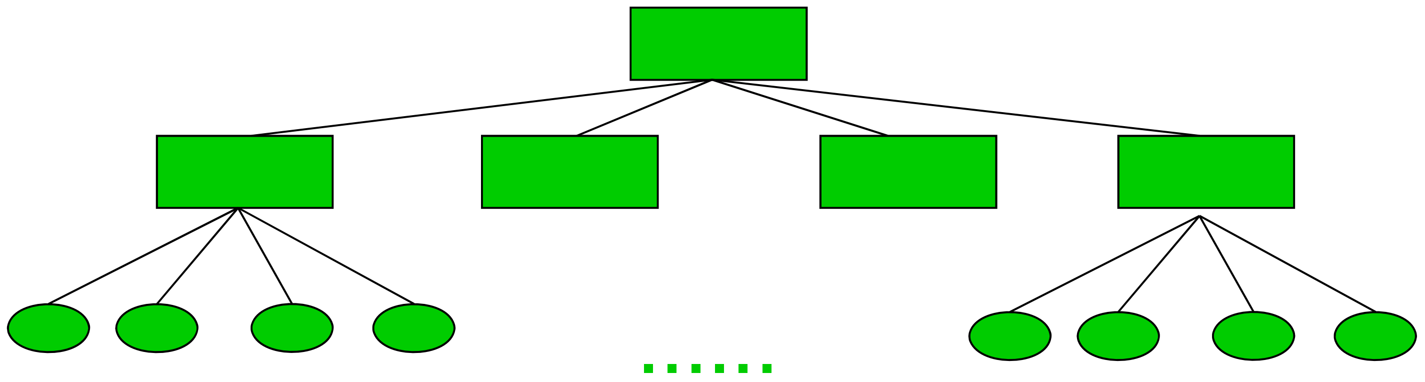
- Memory chip with 4 or 8 DRAM dies over 1 logic die
- Can be placed in an MCM with processor dies
- DRAM dies only store data while logic die handles DRAM control

Future use of logic die:

- Support for Intelligent Memory Operations?
  - Preprocessing data as it is read from memory
  - Performing processor commands "in place"

## Supporting Fine-Grain Parallelism

- Synchronization and communication primitives
  - Efficient point-to-point synch between two cores (F/E bits)
  - Dynamic hierarchical hardware barriers



## Conclusion

- Presented the challenges of Extreme Scale Computing:
  - Designing computers for energy efficiency from the ground up
- Described some of the architecture and design ideas
- We are working to understand and leverage the tradeoff between energy efficiency and resilience

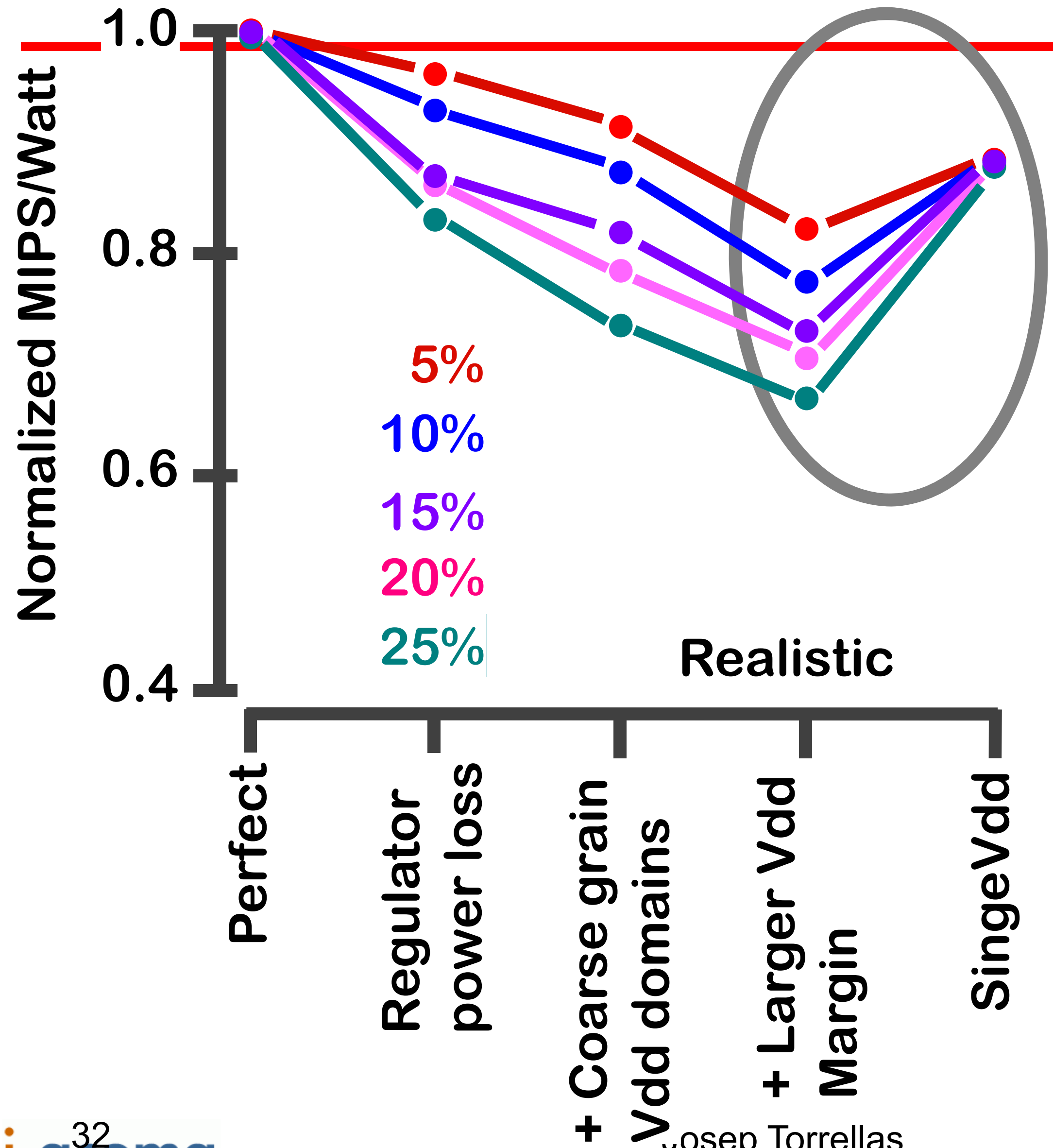
# Extreme Scale Computer Architecture: Energy Efficiency from the Ground Up

**Josep Torrellas**

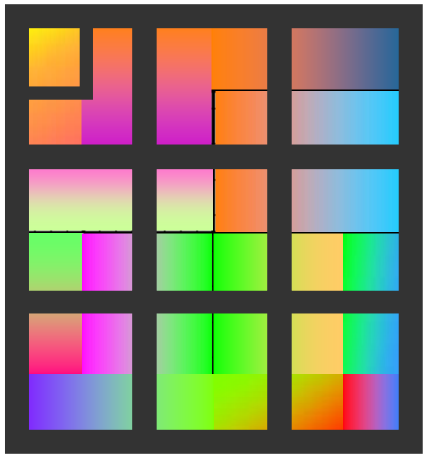
Department of Computer Science
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu

HARSH Workshop, February 2013

# Effectiveness of Single Vdd Domain per Chip



Single Vdd is more E efficient



288-core chip with 8-core clusters