

A Short Tutorial on Thermal Modeling and Management

Kevin Skadron, Mircea Stan, co-PIs

Wei Huang, Karthik Sankaranaryanan

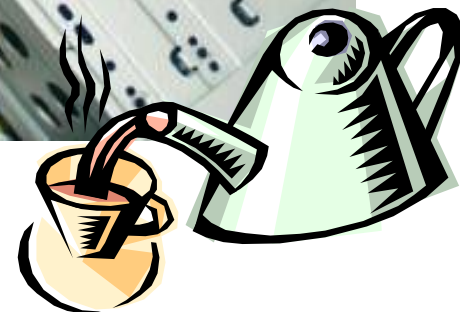
**Univ. of Virginia
HotSpot group**



Cooking-aware computing



- ◆ Some chips rated for 100°C+



Overview

- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Metrics and Design Objectives

- **Power**
 - Average power, instantaneous power, peak power
 - **Energy**
 - Energy (MIPS/W) = heat
 - Energy-Delay product (MIPS²/W)
 - Energy-Delay² product (MIPS³/W) – *voltage indep.*
(Zyuban, GVLIS'02)
 - **Temperature**
 - Correlated with power density over sufficiently large time periods
 - Localized T, short time scales
 - Coarse granularities
- Design for power delivery
- Low-Power Design
- Power-Aware/
Energy-Efficient
Design
- Temperature-Aware Design
- Power-Aware Design
- vs.

Key Differences: Power vs. Thermal

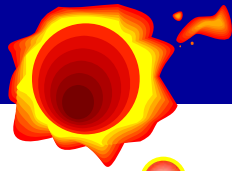
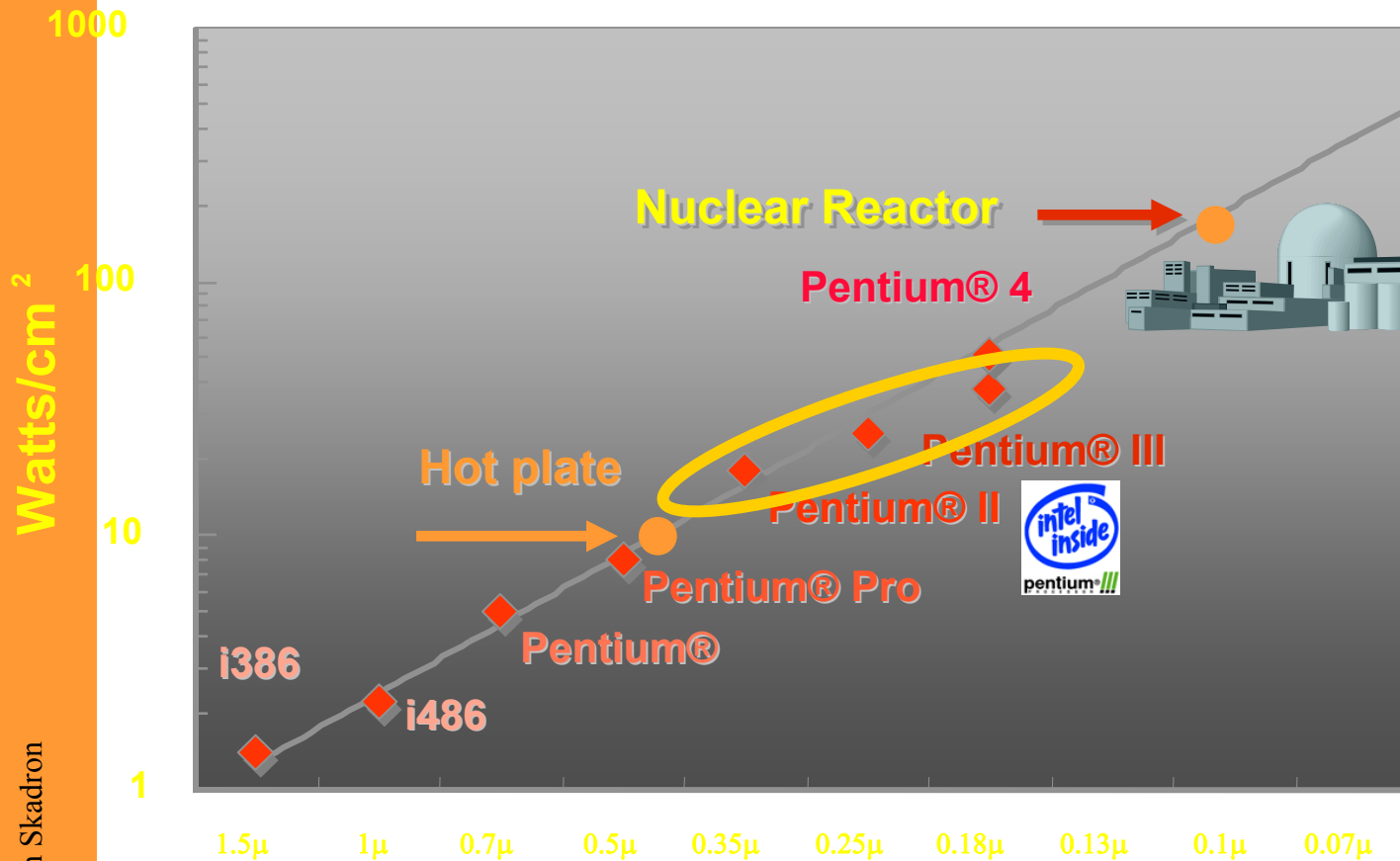
- **Energy efficiency**
 - Reclaim slack
 - Most benefit when system isn't working hard
 - Best effort
- **Thermal**
 - Never exceed max temperature (eg, 100° C)
 - Best effort not sufficient
 - Most important when system is working hard
 - This means that throttling tends to affect performance severely
 - Must provision for worst-case expected workload

Case Study: GPUs

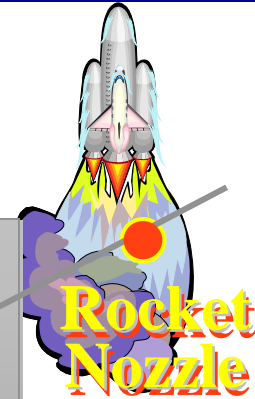
- **For 3D games, frame rate is very important**
- **A board that slows down during the most challenging parts of the game will be unacceptable to gamers**
- **Must provision cooling for most difficult frame of most difficult frame**
- **This means that throttling is only a failsafe**

- **But we want to reduce cooling costs**
- **How?**

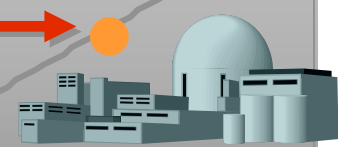
Trends in Power Density



Sun's Surface



Rocket Nozzle



Nuclear Reactor

Hot plate



* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

ITRS Projections

Year	2003	2006	2010	2013	2016
Tech node (nm)	100	70	45	32	22
Vdd (high perf) (V)	1.2	1.1	1.0	0.9	0.8
Vdd (low power) (V)	1.0	0.9	0.7	0.6	0.5
Frequency (high perf) (GHz)	3.0	6.7	15.1	23.0	39.7
	Max power (W)				
High-perf w/ heatsink	149	180	198	198	198
Cost-performance	80	98	119	137	151
Hand-held	2.1	3.0	3.0	3.0	3.0

2001 – was 0.4

ITRS 2006 update

2001 – was 288

- **Clock frequency targets don't account for trend toward simpler cores in multicore**
- **Growth in power *density* means cooling costs continue to grow**
- **High-performance designs seem to be shifting away from clock frequency toward # cores**

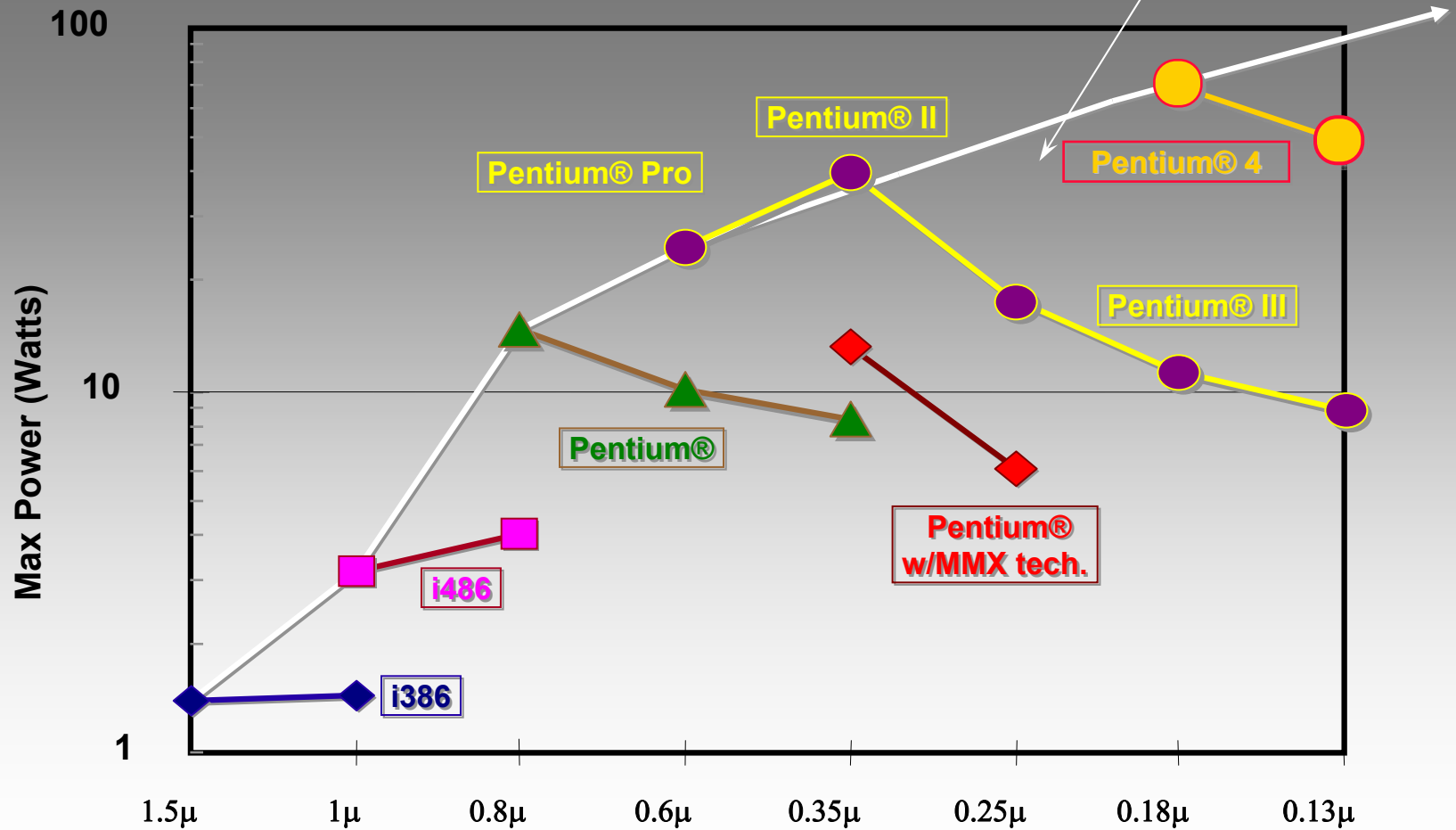
Leakage

- **Vdd reductions were stopped by leakage**
- **Lower Vdd \Rightarrow Vth must be lower**
- **Leakage is exponential in Vth**
- **Leakage is also exponential in T**

Moore's Law and Dennard Scaling

- **Moore's Law: transistor density doubles every N years (currently $N \sim 2$)**
- **Dennard Scaling (constant electric field)**
 - Shrink feature size by k (typ. 0.7), hold electric field constant
 - Area scales by k^2 (1/2) , C , V , delay reduce by k
 - $P \cong CV^2f \Rightarrow P$ goes down by k^2

Actual Power



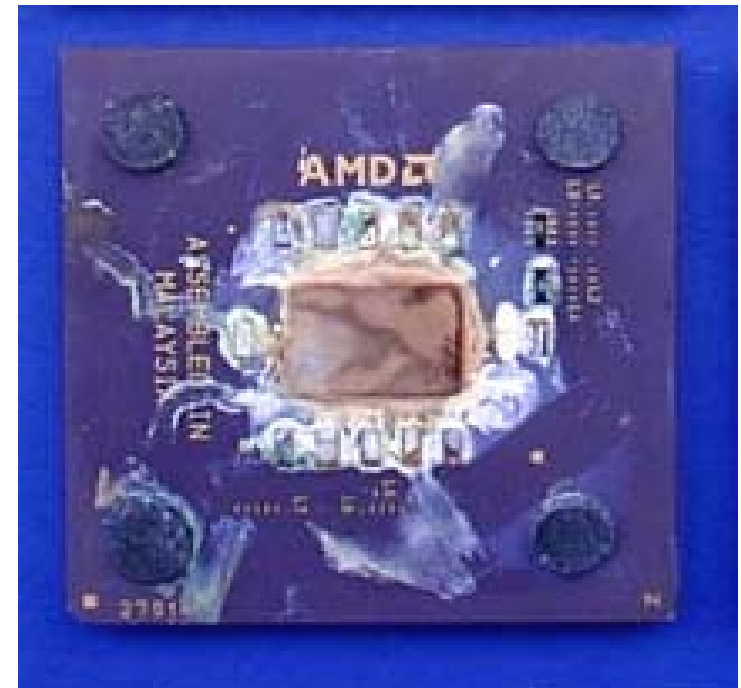
The Real Power Wall

- **Vdd scaling is coming to a halt**
 - Currently 0.9-1.0V, scaling only ~2.5%/gen [ITRS'06]
- **Even if we generously assume C scales and frequency is flat**
 - $P \cong CV^2f \Rightarrow 0.7 (0.975^2) (1) = 0.66$
- **Power *density* goes up**
 - $P/A = 0.66/0.5 = 1.33$
 - And this is very optimistic, because C probably scales more like 0.8 or 0.9, and we want frequency to go up, so a more likely number is **1.5-1.75X**
- **If we keep %-area dedicated to all the cores the same -- total power goes up by same factor**
- **But max TDP for air cooling is expected to stay flat**
- **The shift to multicore *does not eliminate the wall***

ITRS quotes – thermal challenges

- For small dies with high pad count, high power density, or high frequency, “operating temperature, etc for these devices exceed the capabilities of current assembly and packaging technology.”
- “Thermal envelopes imposed by affordable packaging discourage very deep pipelining.”
 - Intel recently canceled its NetBurst microarchitecture
 - Press reports suggest thermal envelopes were a factor

Why we care about thermal issues

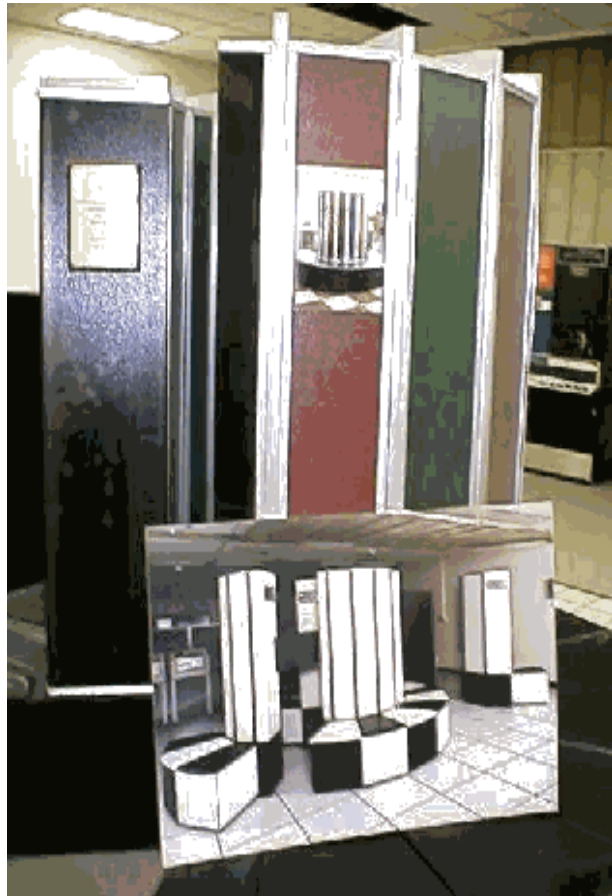


Other Costs of High Heat Flux

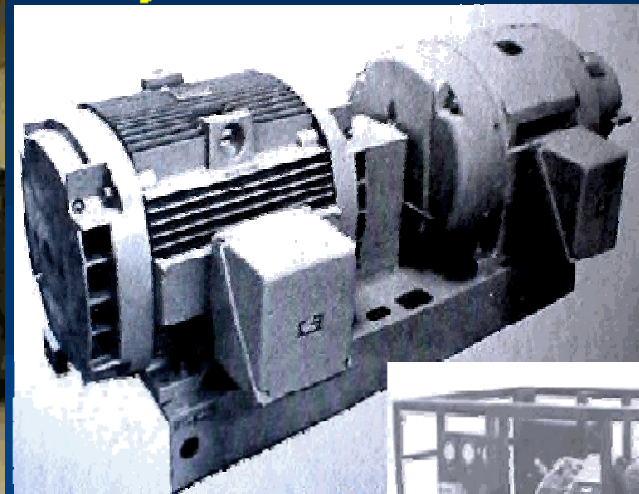
- **Packaging, cooling costs**
- **Noise (quiet high-speed fans are expensive)**
- **Form factors**
- **Some chips may already be underclocked due to thermal constraints!**
 - (especially mobile and sealed systems)
- **Temperature-dependent phenomena**
 - Leakage
 - IR voltage drop (R is T-dep)
 - Aging (e.g. EM)
 - Performance (carrier mobility)

Packaging cost

From Cray (local power generator and refrigeration)...

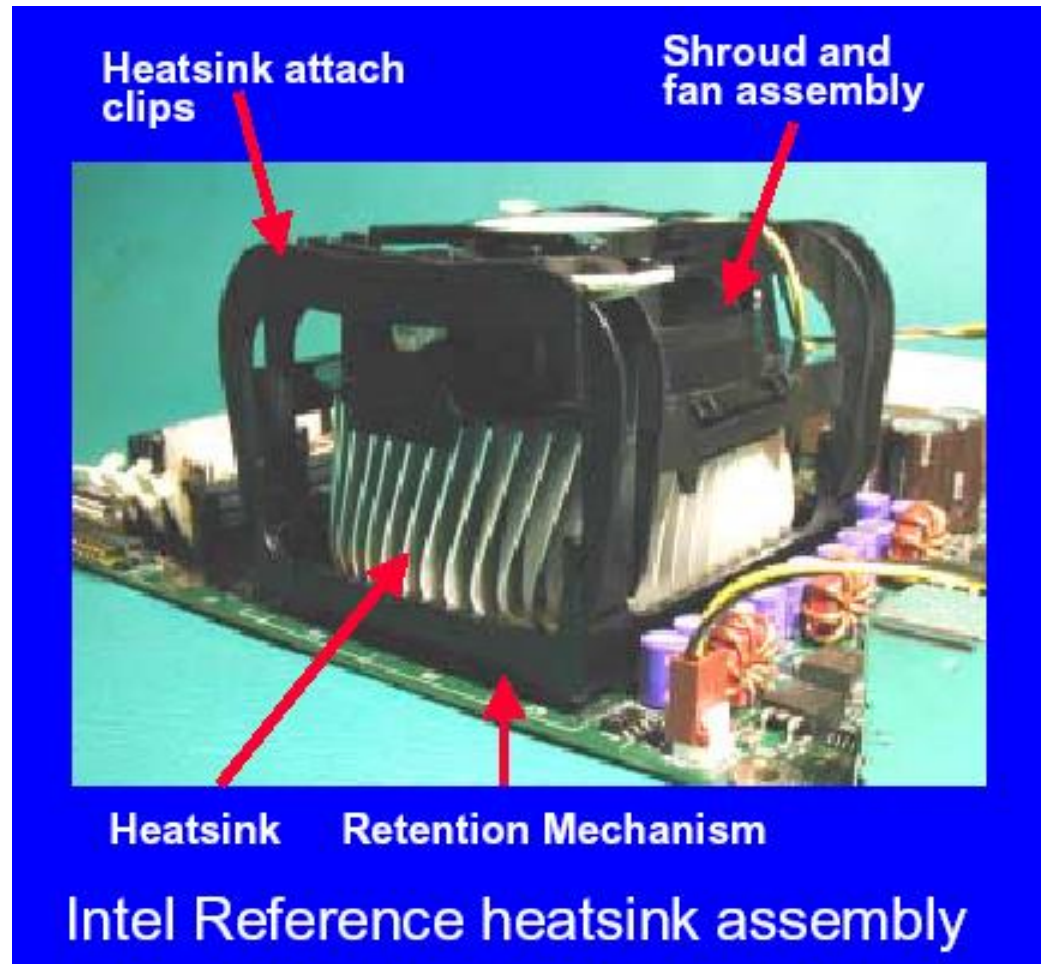


Cray 1 150 Kw. MG set & heat exchanger



Intel Pentium 4 packaging

- **Simpler, but still...**



Graphics Cards

- **Nvidia GeForce 5900 card**



Apple G5 – liquid cooling

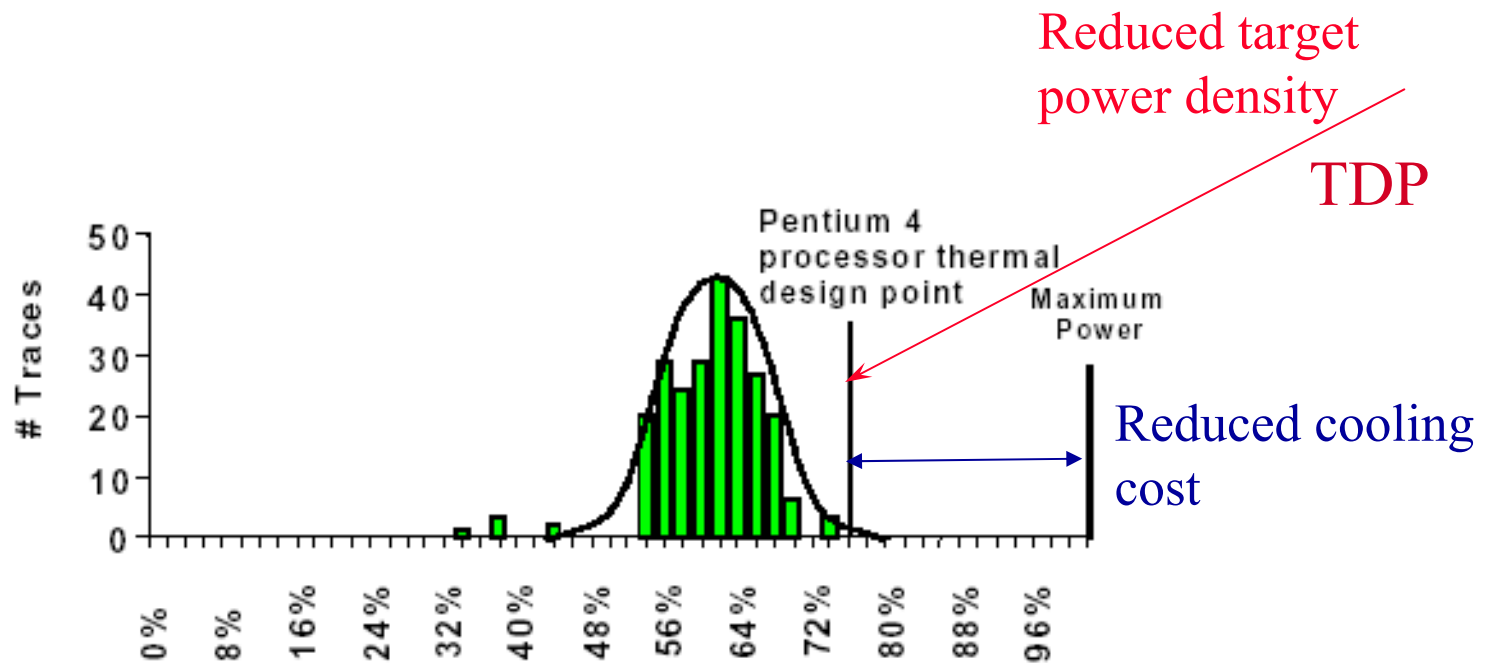
- **Don't know details**
- **In G5 case, liquid is probably for noise**
- **Lots of people in thermal engineering community think liquid is inevitable, especially for server rooms**
- **But others say no:**
 - **This introduces a whole new kind of leakage problem**
 - **Water and electronics don't mix!**

Overview

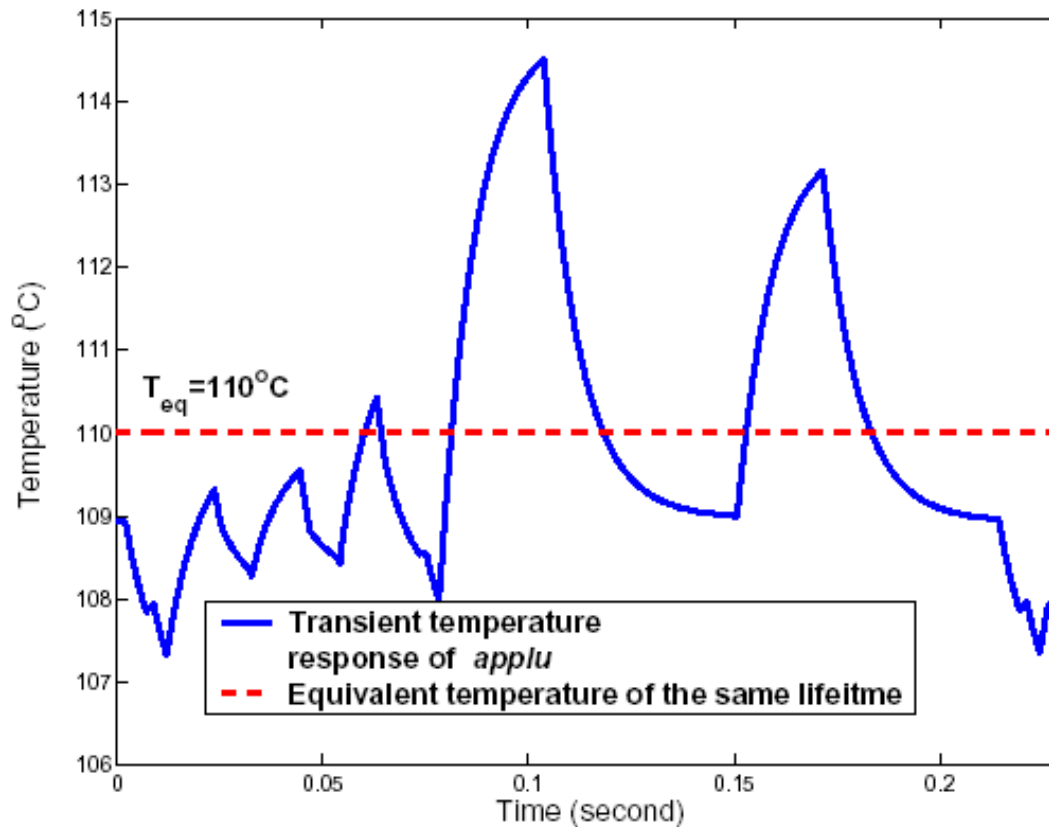
- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Worst-Case leads to Over-design

- **Average case temperature lower than worst-case**
 - Aggressive clock gating
 - Application variations
 - Underutilized resources, e.g. FP units during integer code
- **Currently 20-40% difference**

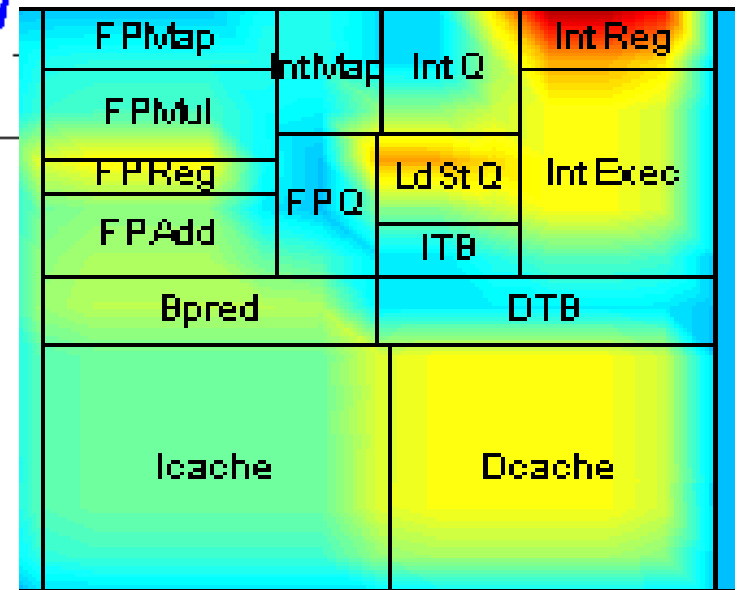


Temporal, Spatial Variations



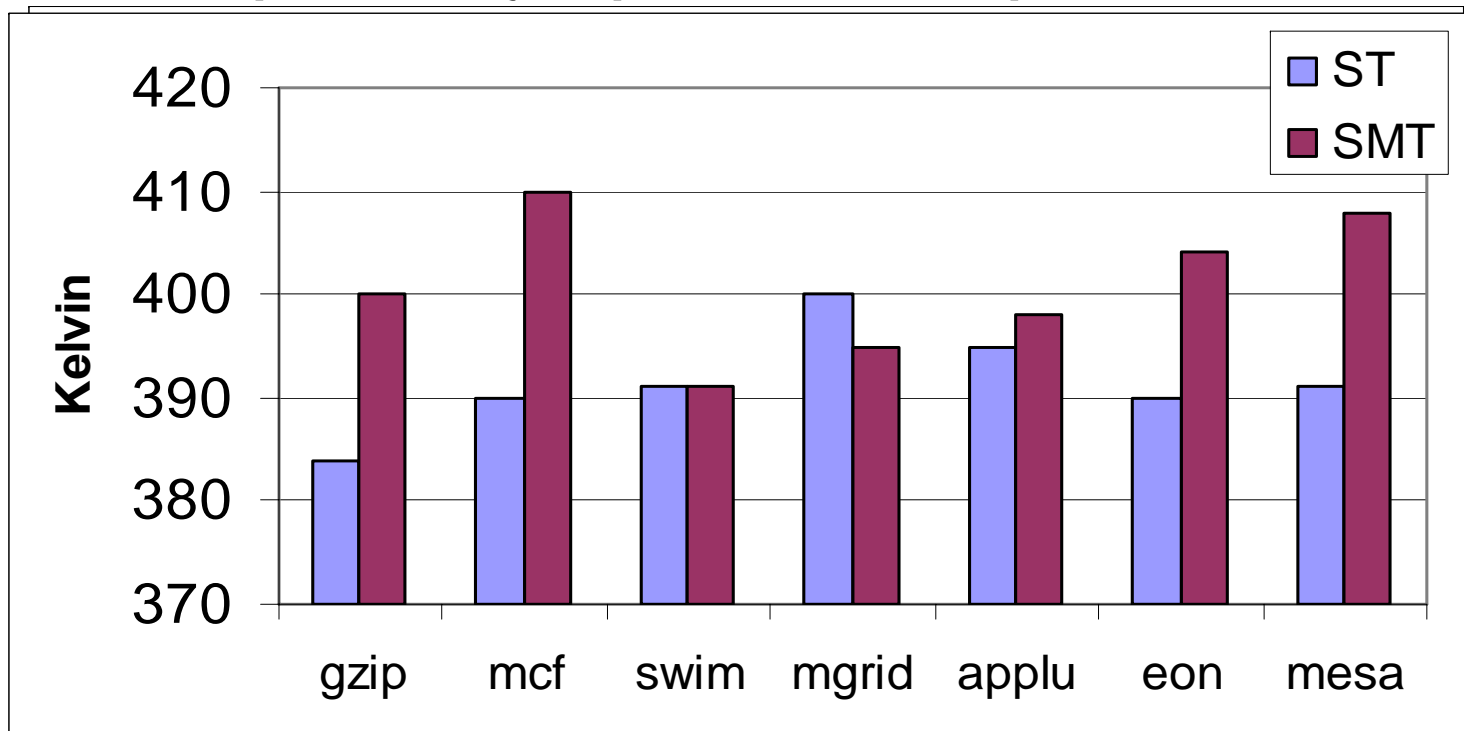
Temperature variation of SPEC applu over time

Hot spots increase cooling costs
 ⇒ must cool for hot spot



Application Variations

- **Wide variation across applications**
- **Architectural and technology trends are making it worse, e.g. *simultaneous multithreading (SMT)***
 - **Leakage is an especially severe problem: *exponentially dependent on temperature!***



Heat vs. Temperature

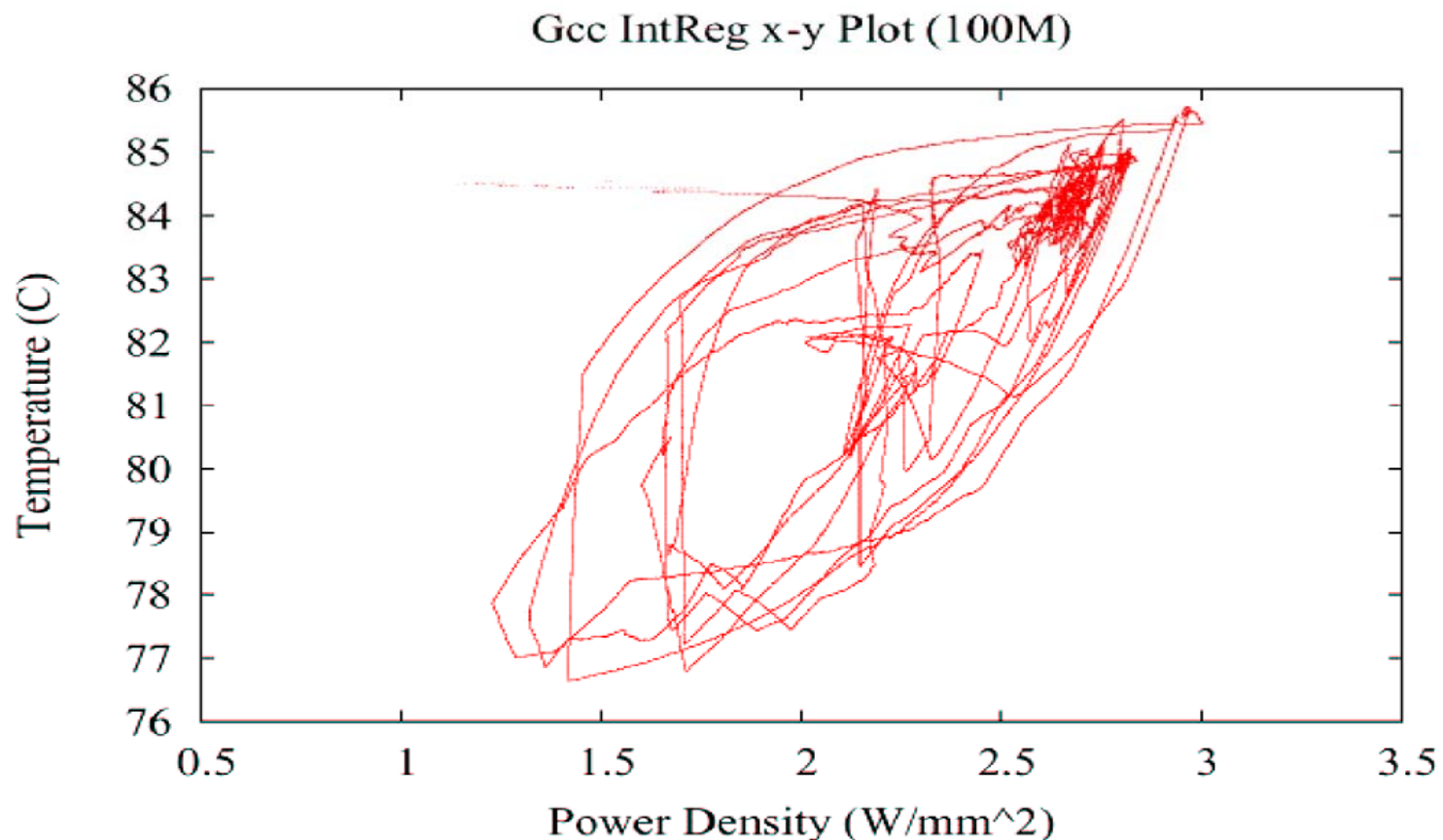
- Different time, space scales
- Heat: no notion of spatial locality

Temperature-aware computing:

Optimize performance subject to a *temperature constraint*

Thermal Modeling: P vs. T

- **Power metrics are an unacceptable proxy**
 - **Chip-wide average won't capture hot spots**
 - **Localized average won't capture lateral coupling**
 - **Different functional units have different power densities**



Thermal consequences

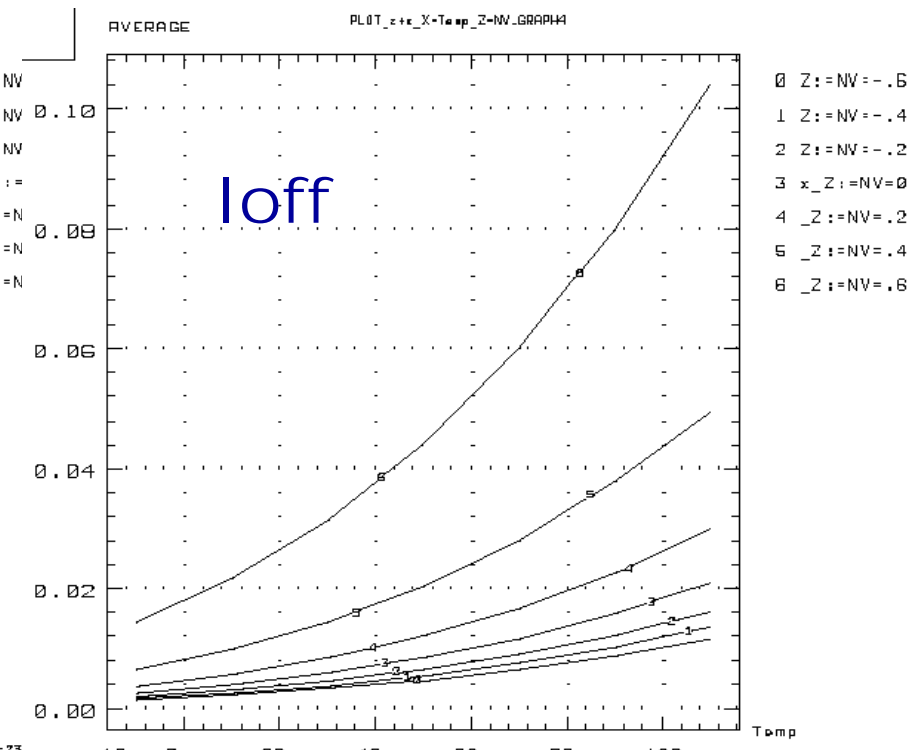
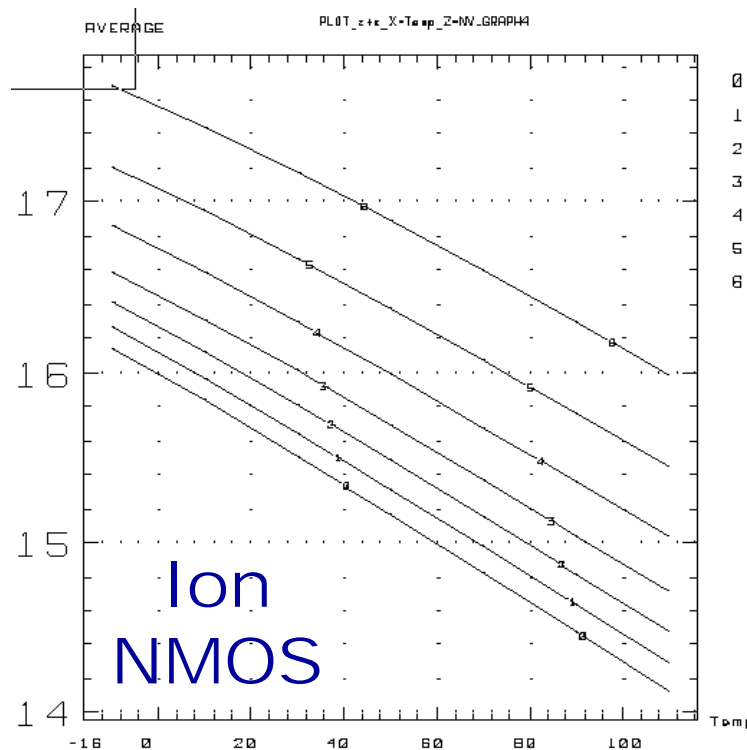
Temperature affects:

- **Circuit performance**
- **Circuit power (leakage)**
- **IC reliability**
- **IC and system packaging cost**
- **Environment**

Performance and leakage

Temperature affects :

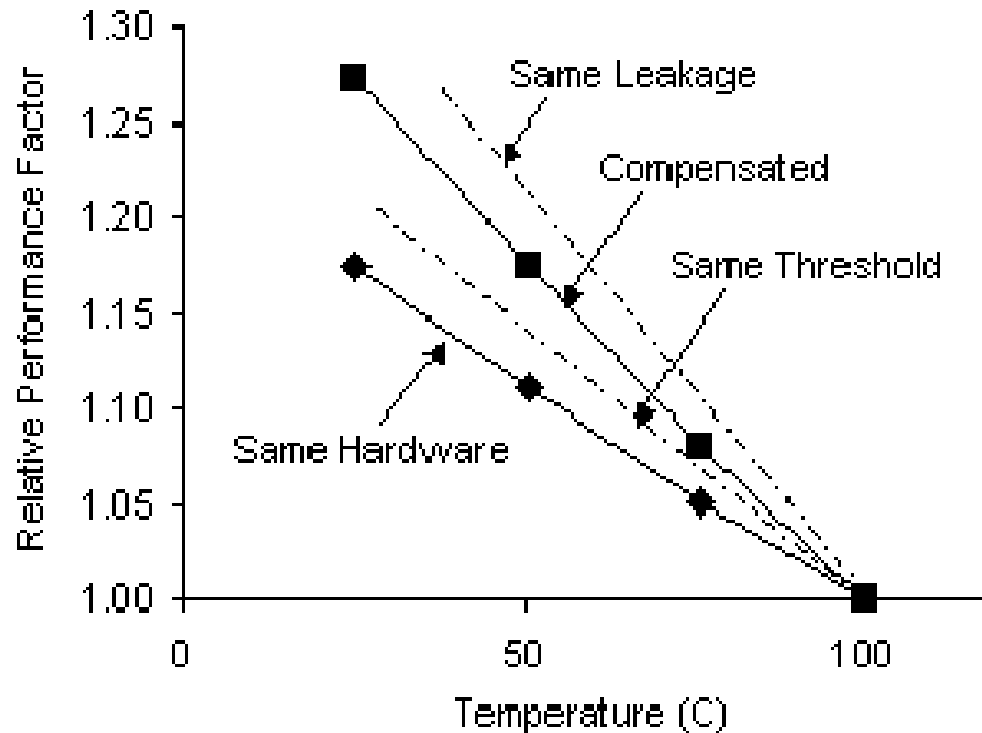
- Transistor threshold and mobility
- Subthreshold leakage, gate leakage
- I_{on} , I_{off} , I_{gate} , delay
- ITRS: 85°C for high-performance, 110°C for embedded!



Temperature-aware circuits

- **Robustness constraint: sets I_{on}/I_{off} ratio**
- **Robustness and reliability: I_{on}/I_{gate} ratio**

Idea: keep ratios constant with T: trade leakage for performance!



Ref: "Ghoshal et al. "Refrigeration Technologies...", ISSCC 2000
Garrett et al. "T3...", ISCAS 2001

Reliability

The Arrhenius Equation: $MTF = A * \exp(E_a / K * T)$

MTF: mean time to failure at T

A: empirical constant

E_a: activation energy

K: Boltzmann's constant

T: absolute temperature

Failure mechanisms:

Die metalization (Corrosion, Electromigration, Contact spiking)

Oxide (charge trapping, gate oxide breakdown, hot electrons)

Device (ionic contamination, second breakdown, surface-charge)

Die attach (fracture, thermal breakdown, adhesion fatigue)

Interconnect (wirebond failure, flip-chip joint failure)

Package (cracking, whisker and dendritic growth, lid seal failure)

Most of the above increase with T (Arrhenius)

Notable exception: hot electrons are worse at low temperatures

More on this later

Overview

- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Heat mechanisms

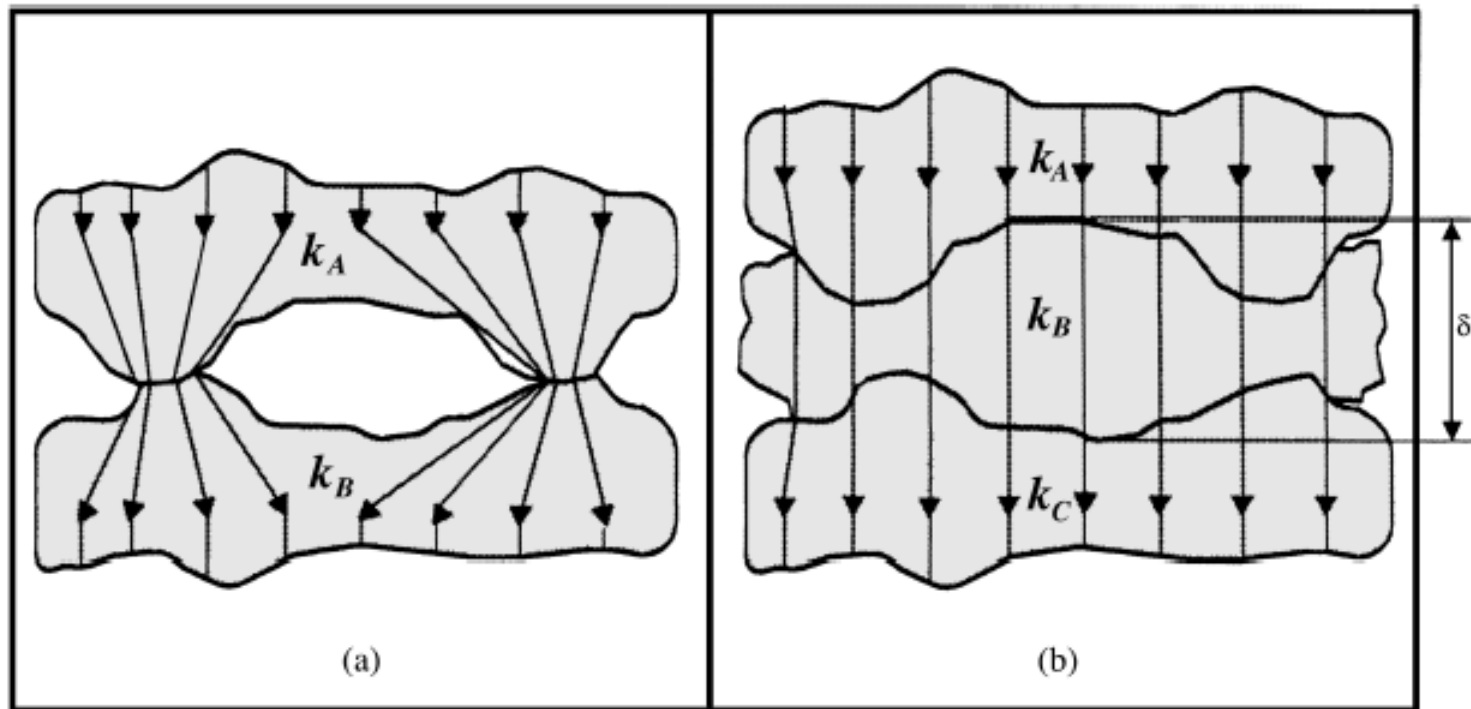
- **Conduction is the main mechanism in a single chip**
 - **Conduction is proportional to the temperature difference and surface area**
- **Convection is the main mechanism in racks, data centers, etc.**

Carnot Efficiency

- Note that in all cases, heat transfer is proportional to ΔT
- This is also one of the reasons energy “harvesting” in computers is probably not cost-effective
 - ΔT w.r.t. ambient is $\ll 100^\circ$
- For example, with a 25W processor, thermoelectric effect yields only $\sim 50\text{mW}$
 - Solbrekken et al, IThERM'04
- This is also why Peltier coolers are not energy efficient
 - 10% eff., vs. 30% for a refrigerator

Surface-to-surface contacts

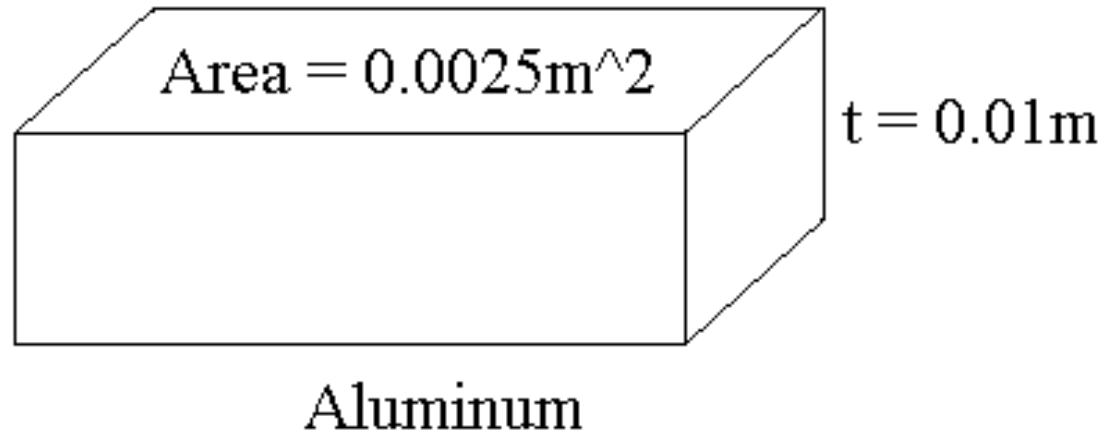
- Not negligible, heat crowding
- Thermal greases/epoxy (can “pump-out”)
- Phase Change Films (undergo a transition from solid to semi-solid with the application of heat)
- Very important to model TIM



Source: CRC Press, R. Rensburg Ed. "Thermal Design of Electronic Equipment", 2001

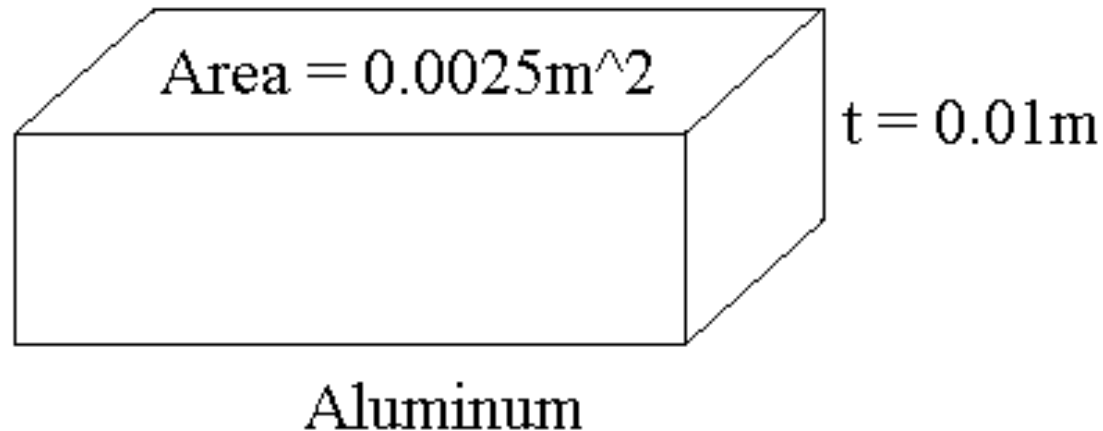
Thermal resistance

- $\Theta = rt / A = t / kA$



Thermal capacitance

- $C_{th} = V \cdot C_p \cdot \rho$



$$\rho(\text{Aluminum}) = 2,710 \text{ kg/m}^3$$

$$C_p(\text{Aluminum}) = 875 \text{ J/(kg}\cdot\text{°C)}$$

$$V = t \cdot A = 0.000025 \text{ m}^3$$

$$C_{bulk} = V \cdot C_p \cdot \rho = 59.28 \text{ J/°C}$$

Simplistic steady-state model

All thermal transfer: $R = k/A$

Power density matters!

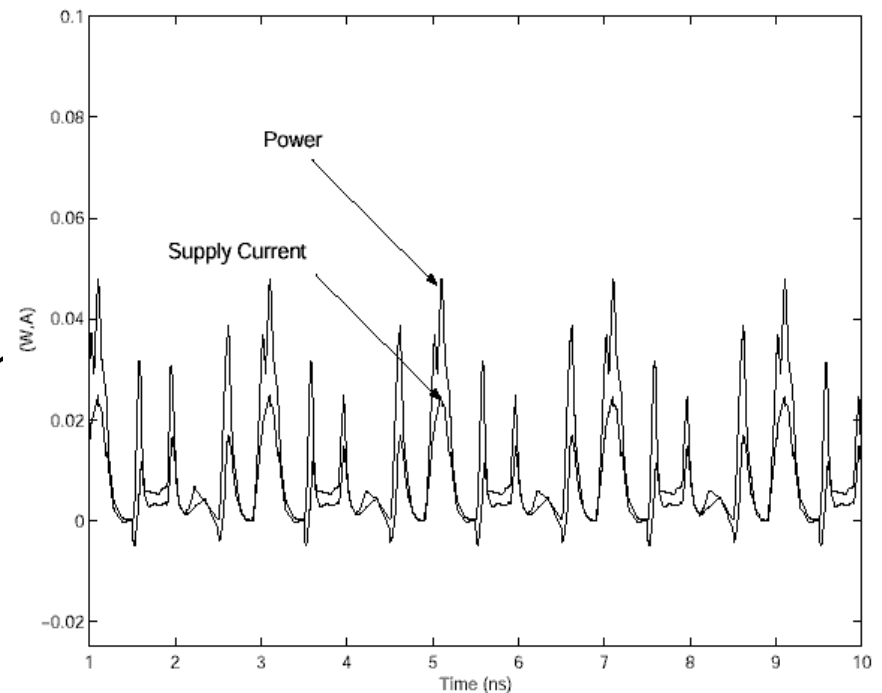
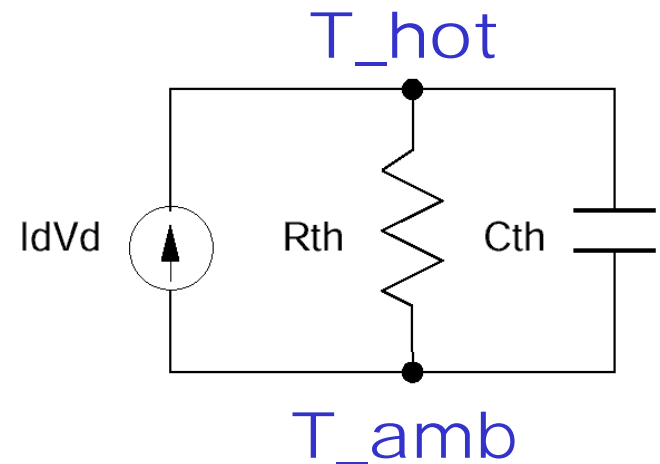
Ohm's law for thermals
(steady-state)

$$\Delta V = I \cdot R \rightarrow \Delta T = P \cdot R$$

$$T_{\text{hot}} = P \cdot R_{\text{th}} + T_{\text{amb}}$$

Ways to reduce T_{hot} :

- reduce P (power-aware)
- reduce R_{th} (packaging, spread I)
- reduce T_{amb} (Alaska?)
- maybe also take advantage of transients (C_{th})



Simplistic dynamic thermal model

Electrical-thermal duality

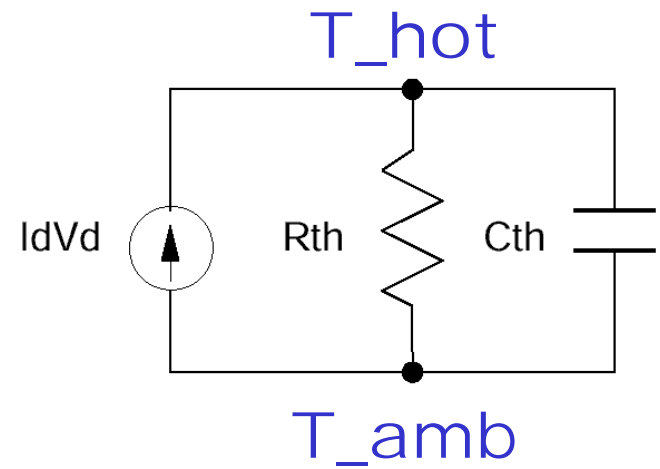
$V \cong \text{temp } (T)$

$I \cong \text{power } (P)$

$R \cong \text{thermal resistance } (R_{th})$

$C \cong \text{thermal capacitance } (C_{th})$

$RC \cong \text{time constant}$



KCL

differential eq. $I = C \cdot dV/dt + V/R$

difference eq. $\Delta V = I/C \cdot \Delta t + V/RC \cdot \Delta t$

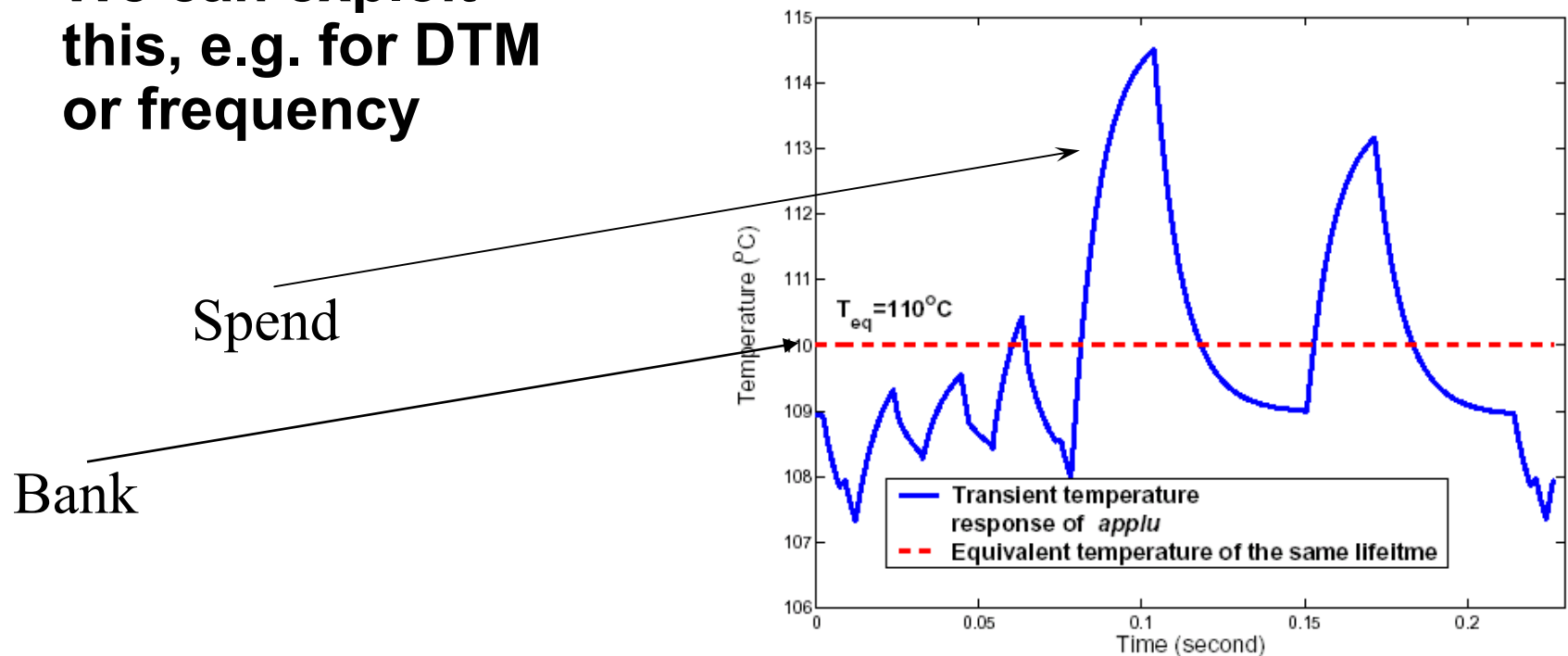
thermal domain $\Delta T = P/C \cdot \Delta t + T/RC \cdot \Delta t$

$(T = T_{hot} - T_{amb})$

One can compute stepwise changes in temperature for any granularity at which one can get P, T, R, C

Reliability as $f(T)$

- Reliability criteria (e.g., DTM thresholds) are typically based on worst-case assumptions
- But actual behavior is often not worst case
- So aging occurs more slowly
- This means the DTM design is over-engineered!
- We can exploit this, e.g. for DTM or frequency



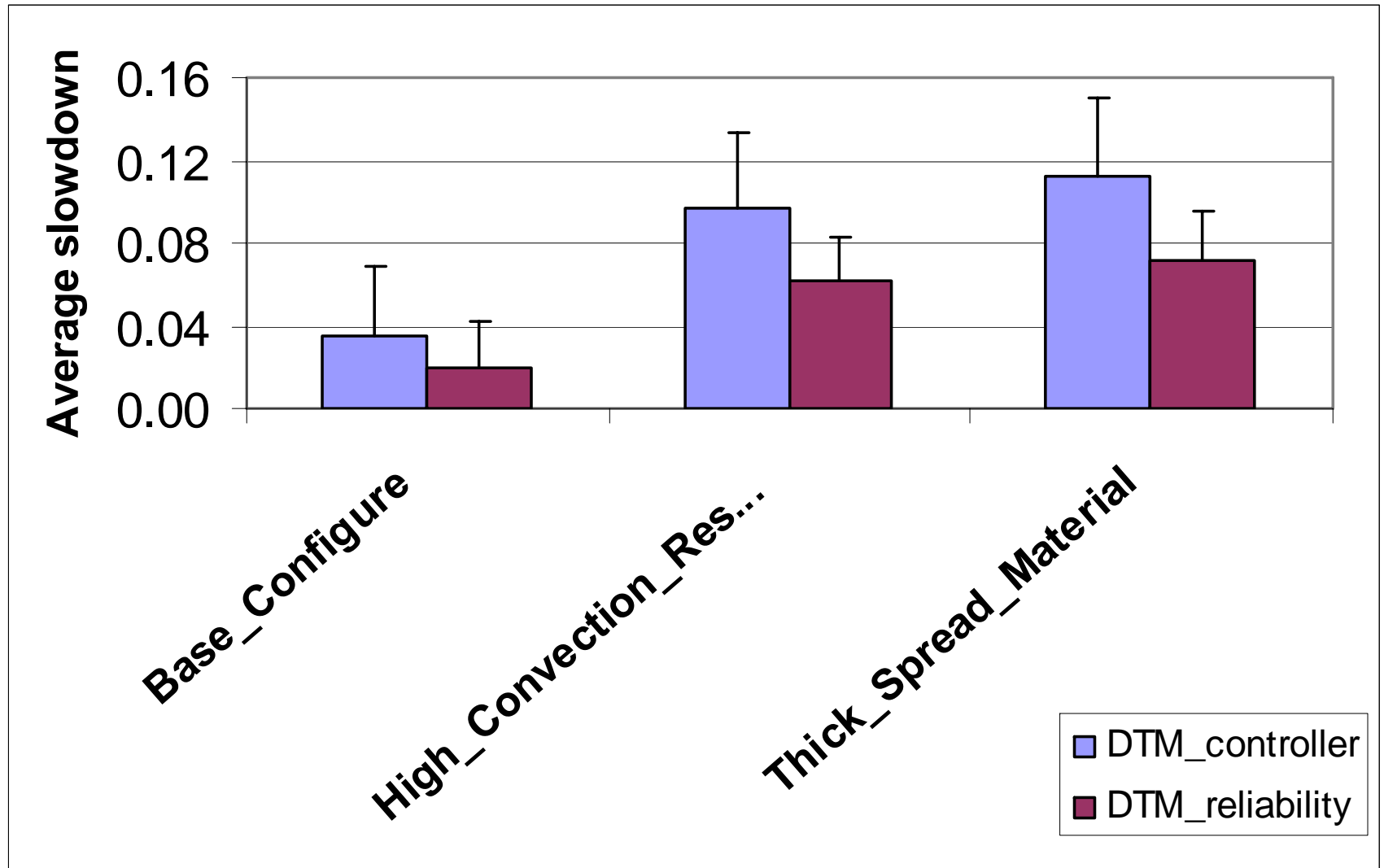
EM Model

$$\int_0^{t_{failure}} \frac{1}{kT(t)} e^{-\frac{E_a}{kT(t)}} dt = \varphi_{th}, \varphi_{th} = const$$

Life Consumption Rate: $R(t) = \frac{1}{kT(t)} e^{-\frac{E_a}{kT(t)}}$

Apply in a “lumped” fashion at the granularity of microarchitecture units, just like RAMP [Srinivasan et al.]

Reliability-Aware DTM



Temperature limits

- **Temperature limits for circuit performance can be measured**
- **Temperature limits for reliability are at best an estimate**
 - **150° is a reasonable rule of thumb for when immediate damage might occur**
 - **Chips are typically specified at lower temperatures, 100-125° for both performance and long-term reliability**
 - **Rule of thumb that every 10° halves circuit lifetime is false**
 - **Originates from a mil-spec that is debunked**
 - **Some reports suggest that it is bump failure, not circuit failure, that really matters**

Thermal issues summary

- **Temperature affects performance, power, and reliability**
- **Architecture-level: conduction only**
 - **Very crude approximation of convection as equivalent resistance**
 - **Convection: too complicated**
 - **Need CFD!**
 - **Radiation: can be ignored**
- **Use compact models for package**
- **Power density is key**
- **Temporal, spatial variation are key**
- **Hot spots drive thermal design**
- **Parameter variations make temperature-aware design even harder (but that's another talk)**

Overview

- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Temperature-Aware Design

- **Worst-case design is wasteful**
- **Power management is not sufficient for chip-level thermal management**
 - **Must target blocks with high power density**
 - **When they are hot**
 - **Spreading heat helps**
 - **Even if energy not affected**
 - **Even if average temperature goes up**
 - **This also helps reduce leakage**

Role of Architecture?

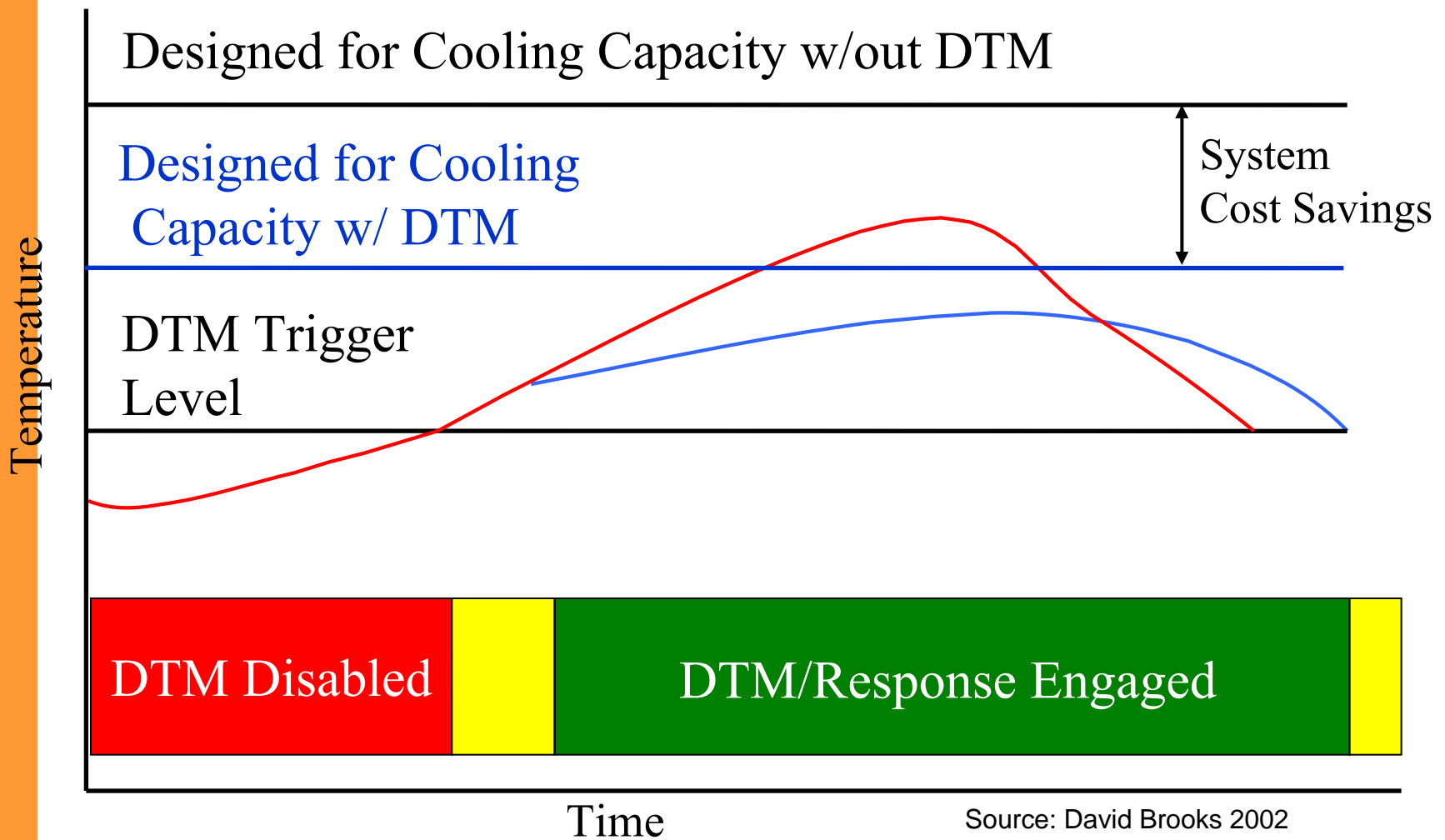
❖ *Temperature-aware architecture*

- Automatic hardware response when temp. exceeds cooling
- Cut power density at runtime, on demand
- Trade reduced costs for occasional performance loss
- Lay out units to maximize thermal uniformity

• **Architecture natural granularity for thermal management**

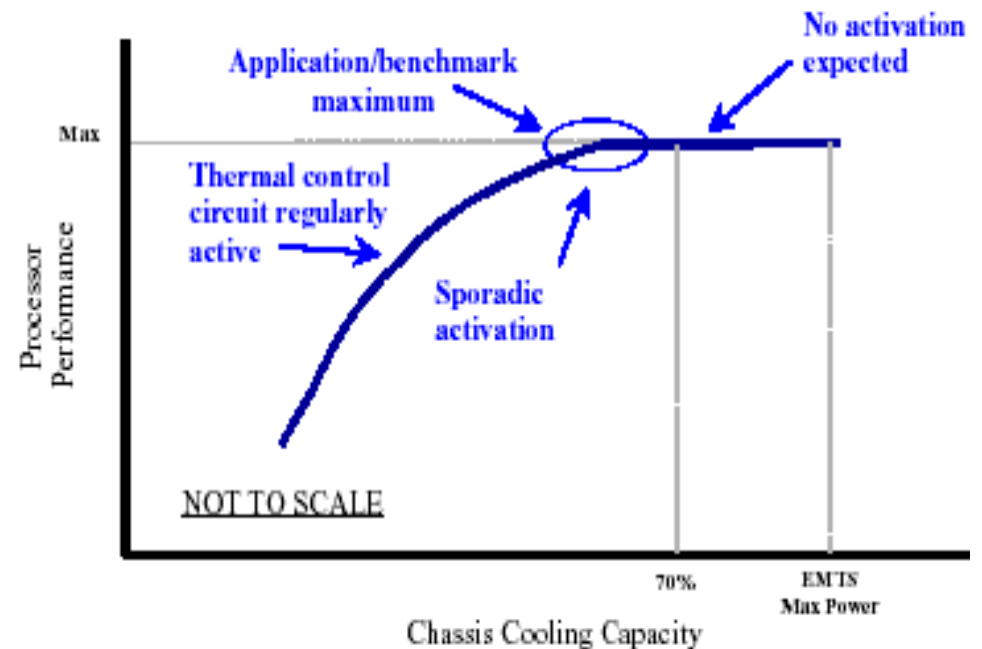
- Activity, temperature correlated within arch. units
- DTM response can target hottest unit: permits fine-tuned response compared to OS or package
- Modern architectures offer rich opportunities for remapping computation
 - e.g., CMPs/SoCs, graphics processors, tiled architectures
 - e.g., register file

Dynamic Thermal Management



DTM

- **Worst case design for the external cooling solution is wasteful**
 - Yet safe temperatures must be maintained when worst case happens
- **Thermal monitors allow**
 - Tradeoff between cost and performance
 - Cheaper package
 - More triggers, less performance
 - Expensive package
 - No triggers full performance

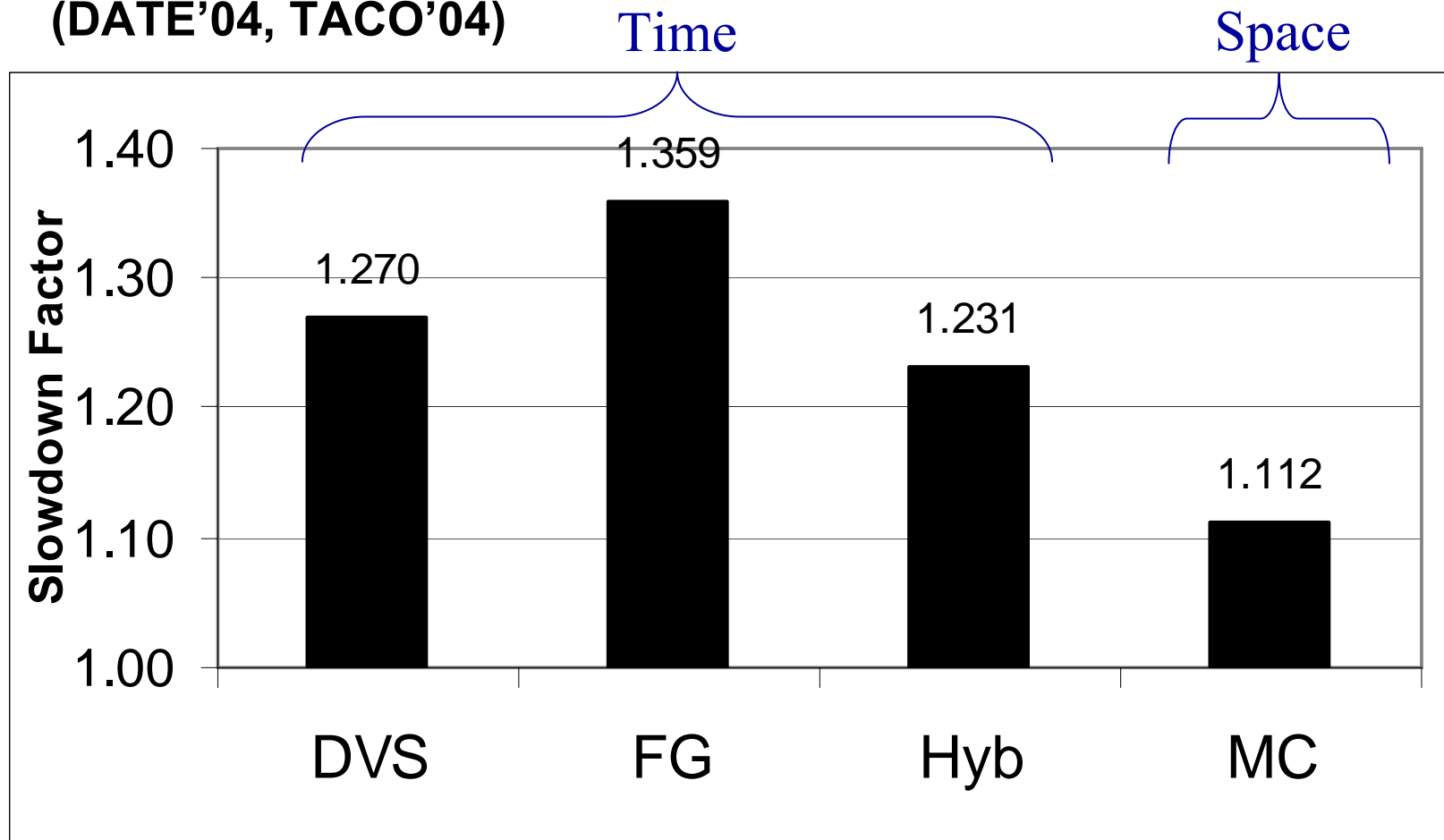


Existing DTM Implementations

- Intel Pentium 4: Global clock gating with shut-down fail-safe
- Intel Pentium M: Dynamic voltage scaling (DVS)
- Intel Core 2: DVS + clock gating + fail-safe
- Transmeta Crusoe: DVS
- IBM Power 5: Probably fetch gating
- ACPI: OS configurable combination of passive & active cooling
- *These solutions sacrifice time (slower or stalled execution) to reduce power density*
- *Better: a solution in “space”*
 - *Tradeoff between exacerbating leakage (more idle logic) or reducing leakage (lower temperatures)*

Space vs. Time

- **Moving the hotspot, rather than throttling it, reduces performance overhead by almost 60%**
- (DATE'04, TACO'04)



**The greater the replication and spread,
the greater the opportunities**

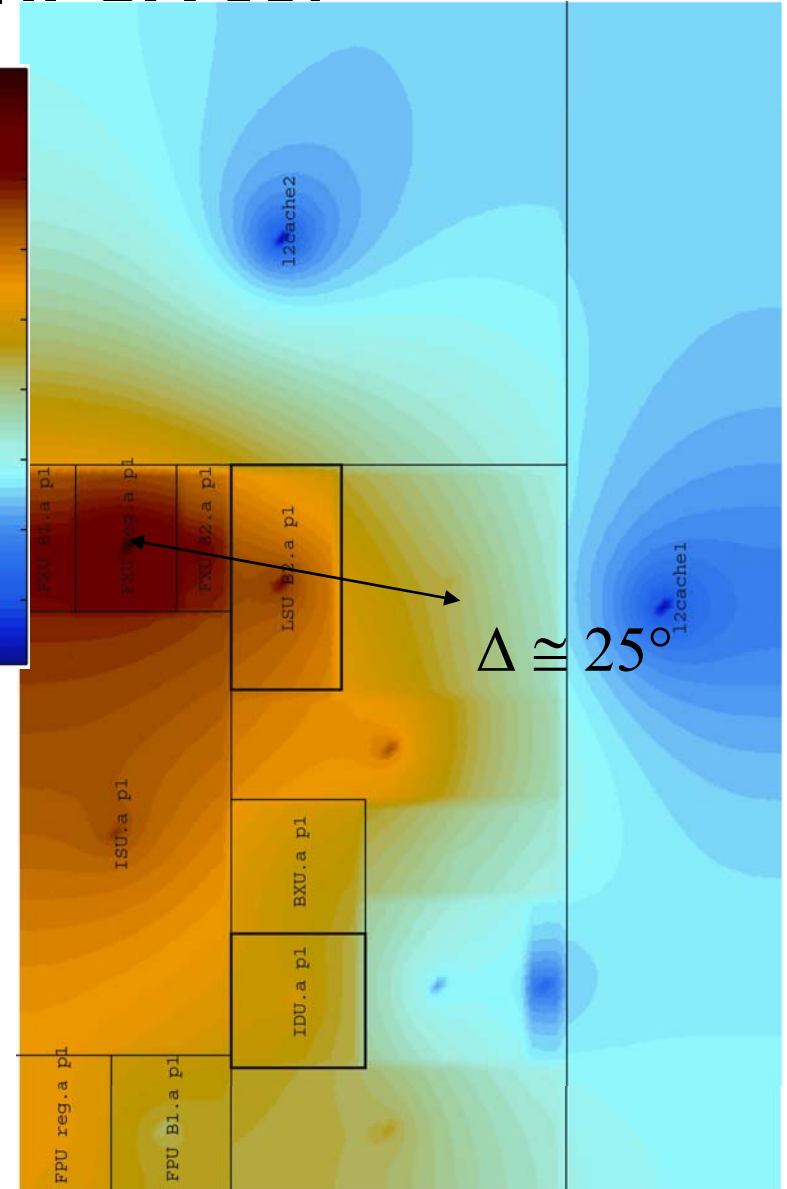
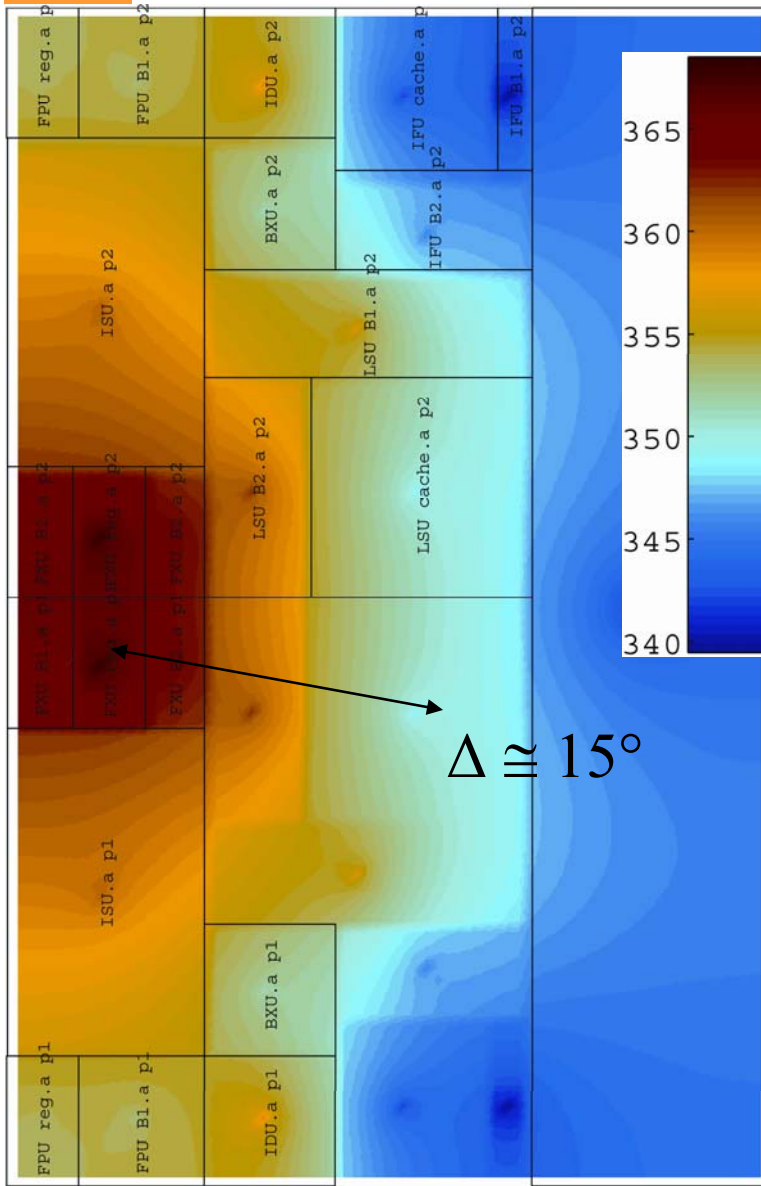
Future DTM considerations

- **Trend in architecture: increasing replication**
 - **Chip multiprocessors**
 - Independent CPUs on a single die
 - Ex: IBM Power5
 - **Tiled organizations**
 - Semi-coupled CPUs
 - Ex: RAW, TRIPS
- **Levels of architectural DTM**
 - **Subunit (single queue entry, register, etc.)**
 - Lots of replication, low migration cost not spread out
 - **Structure (queue, register file, ALU, etc.)**
 - Layout is main lever
 - **Cluster/tile/core**
 - Lots of replication, good spread, but high migration cost, and local hotspots remain

**The greater the replication and spread,
the greater the opportunities**

SMT vs. CMP

- Work w/ Harvard + IBM (HPCA'05)



SMT vs. CMP, cont.

- **CMP is more energy efficient for CPU-bound workloads**
- **SMT can be more energy efficient for memory-bound workloads!**
 - **For same # of threads and equal chip size, CMP has less L2 cache**
- **Localized or hybrid hot-spot management, e.g. intelligent register-file allocation and throttling, can outperform DVS**

Layout Considerations

- **Multicore layout and “spatial filtering” give you an extra lever (DAC’08, to appear)**
 - The smaller a power dissipator, the more effectively it spreads its heat [IEEE Trans. Computers, to appear]
 - Ex: 2x2 grid vs. 21x21 grid: 188W TDP vs. 220 W (17%) – DAC 2008
 - Increase core density
 - Or raise Vdd, Vth, etc.
 - Thinner dies, better packaging boost this effect
- **Seek architectures that minimize area of high power density, maximize area in between, *and can be easily partitioned***



Overview

- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Thermal modeling

- **Want a fine-grained, dynamic model of *temperature***
 - At a granularity architects can reason about
 - That accounts for adjacency and package
 - That does not require detailed designs
 - That is fast enough for practical use
- **HotSpot - a compact model based on thermal R, C (HPCA'02, ISCA'03)**
 - Parameterized to automatically derive a model based on various
 - Architectures
 - Power models
 - Floorplans
 - Thermal Packages

Dynamic compact thermal model

Electrical-thermal duality

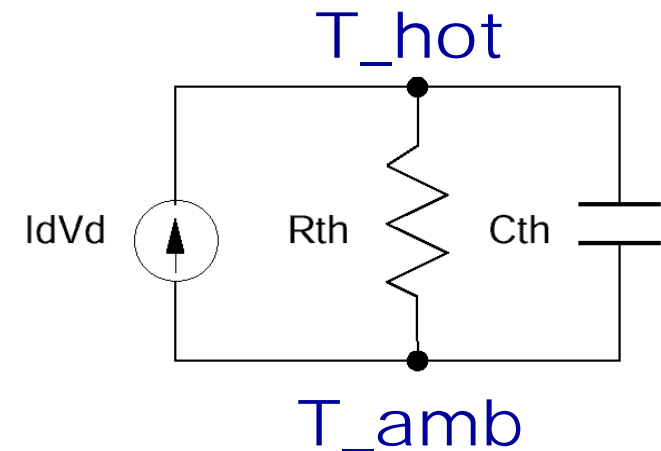
$V \cong \text{temp (T)}$

$I \cong \text{power (P)}$

$R \cong \text{thermal resistance (Rth)}$

$C \cong \text{thermal capacitance (Cth)}$

RC time constant (Rth Cth)



Kirchoff Current Law

differential eq. $I = C \cdot dV/dt + V/R$

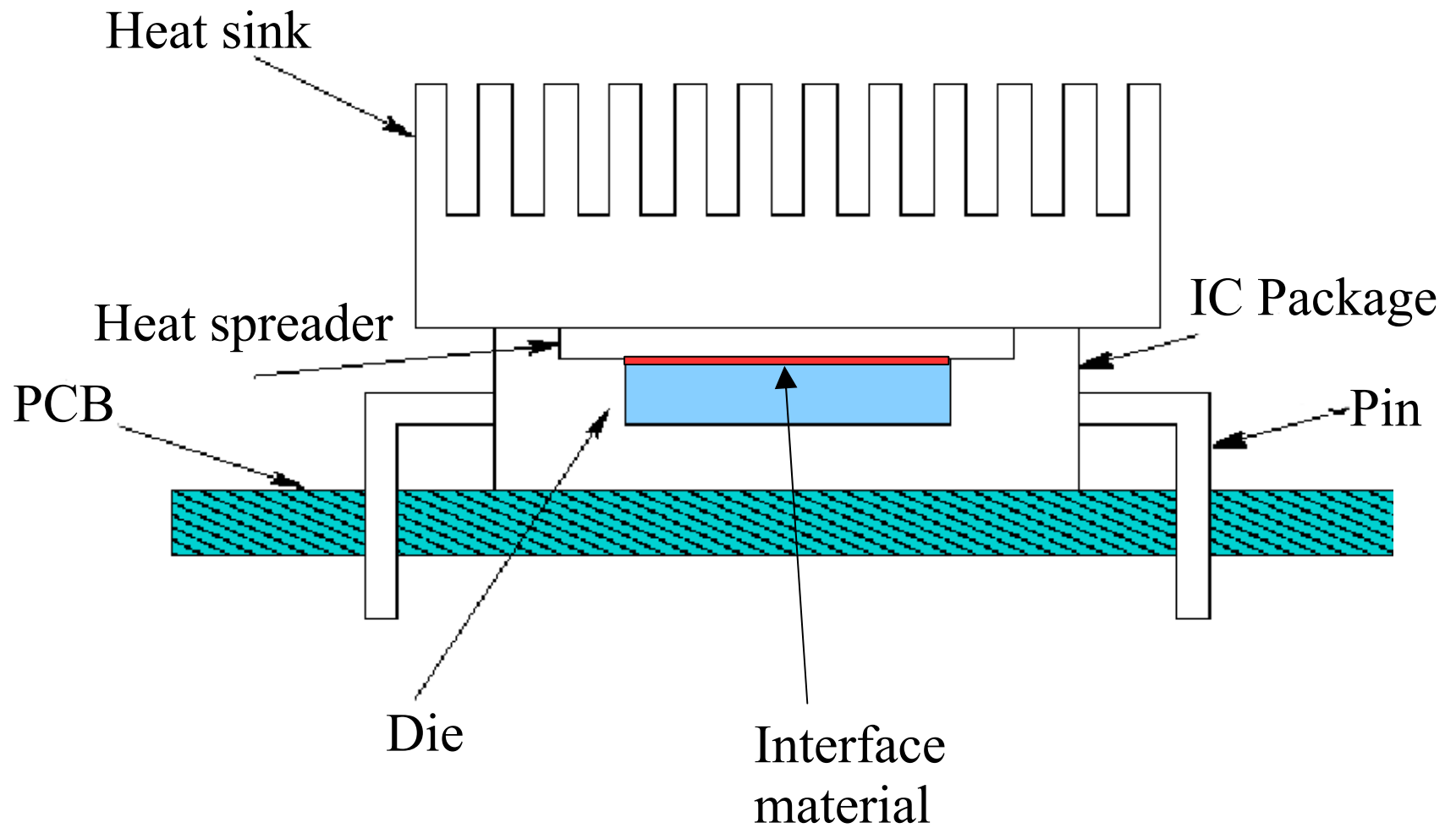
thermal domain $P = C_{th} \cdot dT/dt + T/R_{th}$

where $T = T_{hot} - T_{amb}$

At higher granularities of P, Rth, Cth

P, T are vectors and Rth, Cth are circuit matrices

Example System

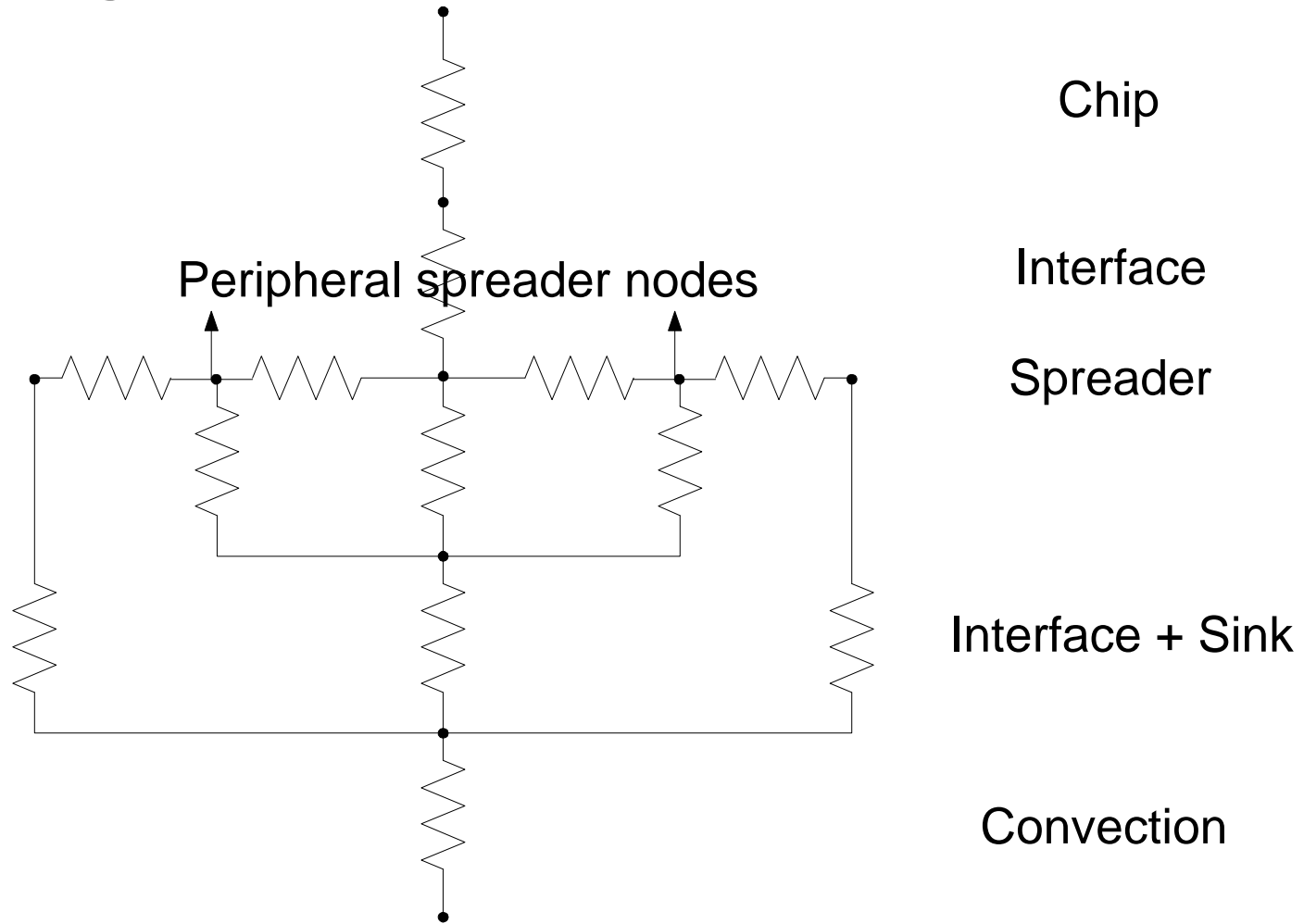


Modeling the package

- **Thermal management allows for packaging alternatives/shortcuts/interactions**
- **HotSpot needs a model of packaging**
- **Basic thermal model:**
 - Heat spreader
 - Heatsink
 - Interface materials (e.g. epoxy)
 - Fan/Active cooler
- **Thermal resistance due to convection**
- **Constriction and bulk resistance for fins**
- **Spreading constriction and bulk resistance for heatsink base and heat spreader**
- **Thermal resistance for interface materials**
- **Thermal capacitance heat spreader and heatsink**

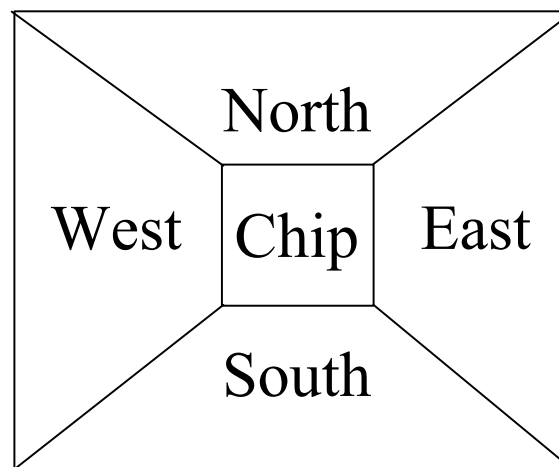
Equivalent vertical network

- **Diagram is simplified – peripheral nodes**



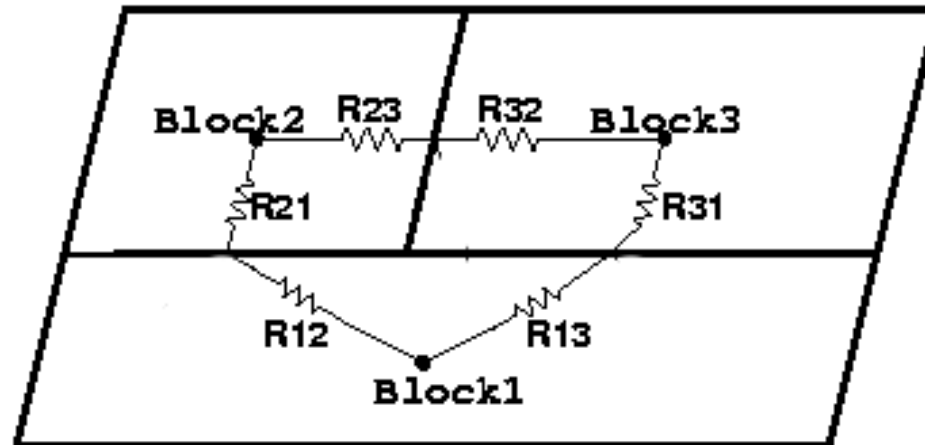
Vertical network parameters

- **Resistances**
 - Determined by the corresponding areas and their cross sectional thickness
 - $R = \text{resistivity} \times \text{thickness} / \text{Area}$
- **Capacitances**
 - $C = \text{specific heat} \times \text{thickness} \times \text{Area}$
- **Peripheral node areas**



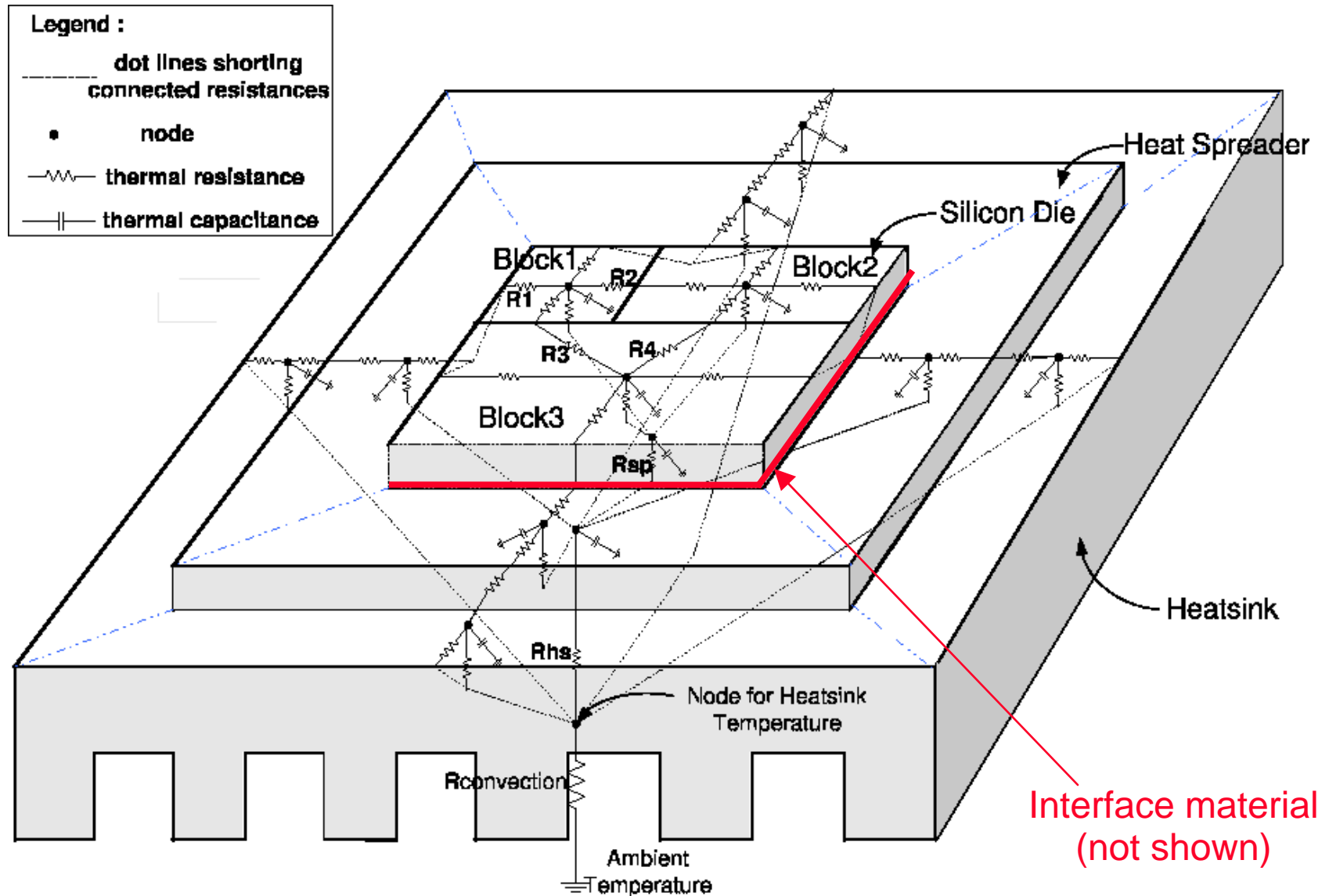
Spreader

Lateral resistances



- **Determined by the floorplan and the length of shared edges between adjacent blocks**

Our model (lateral and vertical)



Temperature equations

- **Fundamental RC differential equation**
 - $P = C \, dT/dt + T / R$
- **Steady state**
 - $dT/dt = 0$
 - $P = T / R$
- **When R and C are network matrices**
 - **Steady state – $T = R \times P$**
 - **Modified transient equation**
 - $dT/dt + (RC)^{-1} \times T = C^{-1} \times P$
 - **HotSpot software mainly solves these two equations**

HotSpot

- **Time evolution of temperature is driven by unit activities and power dissipations averaged over 10K cycles**
 - **Power dissipations can come from any power simulator, act as “current sources” in RC circuit ('P' vector in the equations)**
 - **Simulation overhead in Wattch/SimpleScalar: < 1%**
- **Requires models of**
 - **Floorplan: *important for adjacency***
 - **Package: *important for spreading and time constants***
 - ***R and C matrices are derived from the above***

Implementation

- **Primarily a circuit solver**
- **Steady state solution**
 - **Mainly matrix inversion – done in two steps**
 - **Decomposition of the matrix into lower and upper triangular matrices**
 - **Successive backward substitution of solved variables**
 - **Implements the pseudocode from CLR**
- **Transient solution**
 - **Inputs – current temperature and power**
 - **Output – temperature for the next interval**
 - **Computed using a fourth order Runge-Kutta (RK4) method**

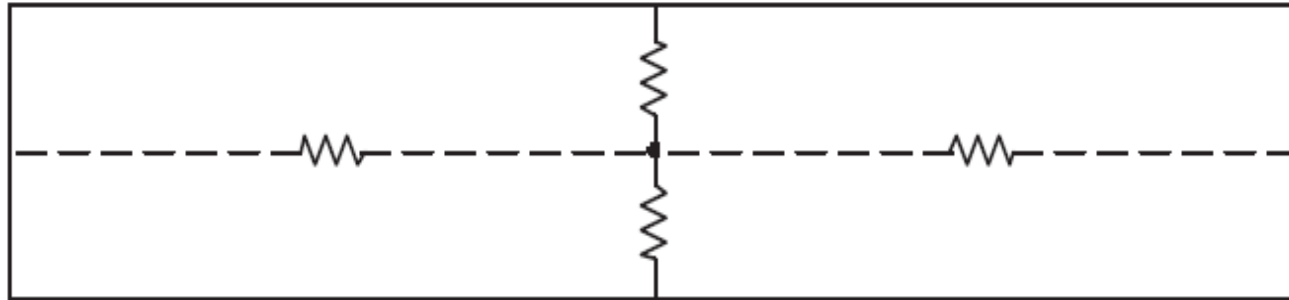
Transient solution

- **Solves differential equations of the form $dT + AT = B$ where A and B are constants**
 - In HotSpot, A is constant (RC) but B depends on the power dissipation
 - Solution – assume constant average power dissipation within an interval (10 K cycles) and call RK4 at the end of each interval
- **In RK4, current temperature (at t) is advanced in very small steps ($t+h$, $t+2h$...) till the next interval (10K cycles)**
 - Step size determined adaptively to minimize overhead, maximize speed of convergence
- **RK – `4` because error term is 4th order i.e., $O(h^4)$**

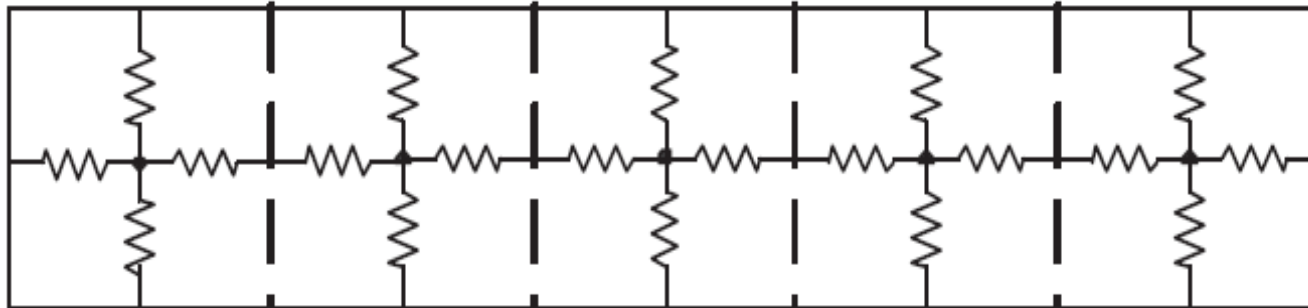
Transient solution contd...

- **4th order error has to be within the required precision**
- **The step size (h) has to be small enough even for the maximum slope of the temperature evolution curve**
- **Transient solution for the differential equation is of the form Ae^{-Bt} with A and B are dependent on the RC network**
- **Thus, the maximum value of the slope (AxB) and the step size are computed accordingly**

Block sub-division



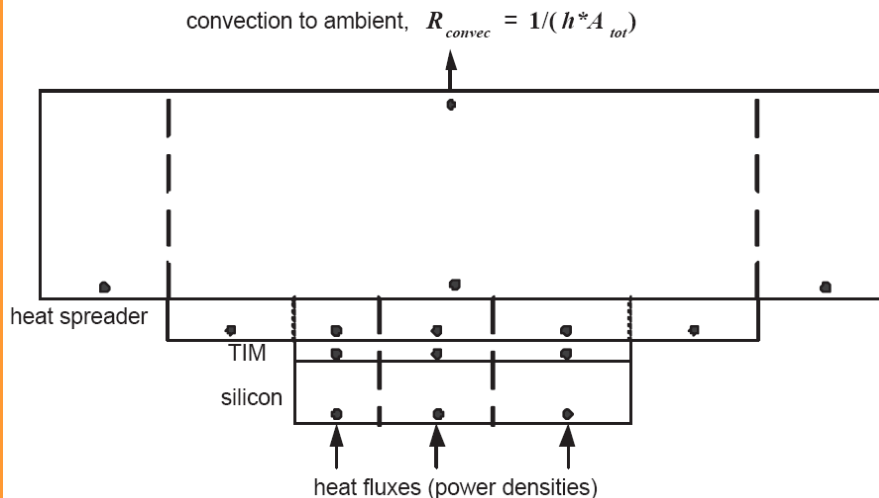
Version 3.1 – a block is represented by a single node



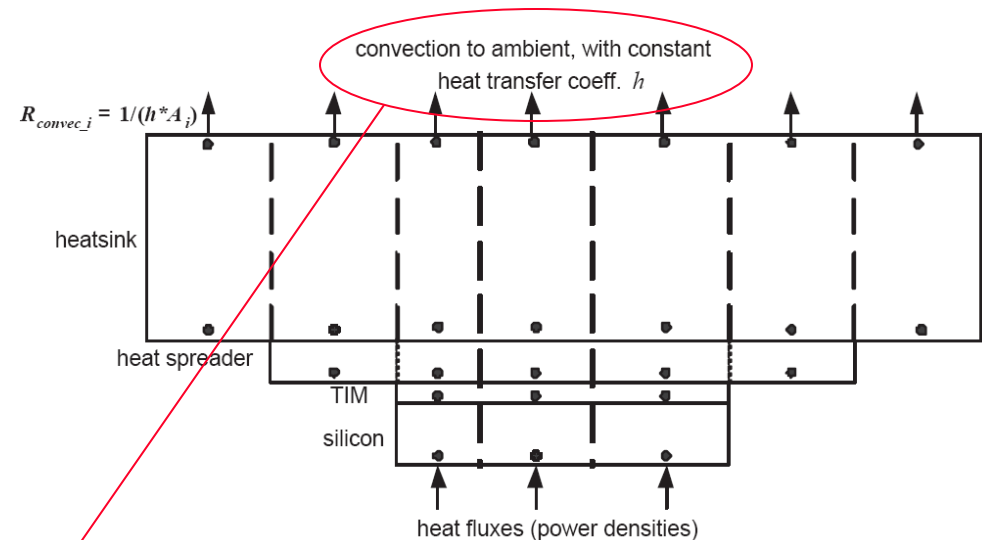
Version 4.0 – sub-blocks with aspect ratio close to 1

Heat sink boundary condition

- Accuracy improvements in v 4.0 (WDDD'07)



Version 3.1 – single convection resistance, isothermal surface



Version 4.0 – parallel convection resistances, center modeled at the same level of detail as silicon

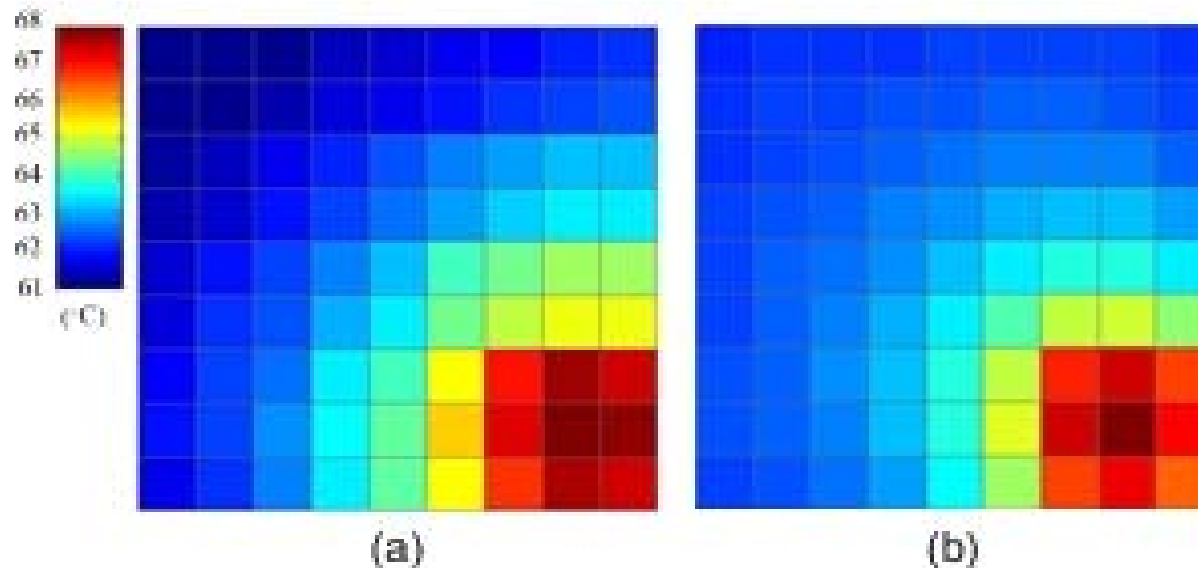
- Can also model systems with no heat sink

HotSpot

- **First crude model developed in 2001**
- **Version 1 released in 2003**
- **Version 4.1 just released**
- **Over 1400 downloads, over 550 citations of HotSpot papers (according to Google Scholar)**
- **Most recent improvements, analysis to appear in *IEEE Trans. Computers* (preprint should be online soon)**
- **HotSpot also includes:**
 - **grid model (using multigrid solution)**
 - **floorplanning tools**
- **<http://lava.cs.virginia.edu/HotSpot>**

Validation (1)

- **First validated and calibrated using MICRED test chips (see DAC'04 paper)**
 - 9x9 array of power dissipators and sensors
 - Compared to HotSpot configured with same grid, package



- **Within 7% for both steady-state and transient step-response**
 - Interface material (chip/spreader) matters

Validation (2)

- **POWER5 ANSYS model**
- **FPGA (ICCD 2005)**
- **Infrared measurements, in collaboration with Jose Renau (using methodology in his ISCA'07 paper)**

Notes

- **Note that HotSpot currently measures temperatures *in the silicon***
 - But that's also what the most sensors measure
- **Temperature continues to rise through each layer of the die**
 - Temperature in upper-level metal is considerably higher
 - Interconnect model released soon!
- **Time constants in package much higher than in silicon**

Soon to be features

- **Temperature models for wires, pads and interface material between heat sink and spreader**
 - See DAC'04 paper
- **Interface for package selection**
- **Excel interface**
- **Better integration with leakage modeling**

Overview

- 1. What is thermal-aware design?**
- 2. Why thermal?**
- 3. Some basic heat transfer concepts**
- 4. Thermal management**
- 5. HotSpot thermal model**
- 6. Thermal sensor issues**

Sensors

Caveat emptor:

We are not well-versed on sensor design; the following is a digest of information we have been able to collect from industry sources and the research literature.

Desirable Sensor Characteristics

- **Small area**
- **Low Power**
- **High Accuracy + Linearity**
- **Easy access and low access time**
- **Fast response time (*slew rate*)**
- **Easy calibration**
- **Low sensitivity to process and supply noise**

Types of Sensors

(In approx. order of increasing ease to build)

- **Thermocouples – voltage output**
 - Junction between wires of different materials; voltage at terminals is $\propto T_{\text{ref}} - T_{\text{junction}}$
 - Often used for external measurements
- **Thermal diodes – voltage output**
 - Biased p-n junction; voltage drop for a known current is temperature-dependent
- **Biased resistors (*thermistors*) – voltage output**
 - Voltage drop for a known current is temperature dependent
 - You can also think of this as varying R
 - Example: 1 K Ω metal “snake”
- **BiCMOS, CMOS – voltage or current output**
 - Rely on reference voltage or current generated from a reference band-gap circuit; current-based designs often depend on temperature-dependence of threshold
- **4T RAM cell – decay time is temp-dependent**
 - [Kaxiras et al, ISLPED'04]

Sensors: Problem Issues

- **Poor control of CMOS transistor parameters**
- **Noisy environment**
 - **Cross talk**
 - **Ground noise**
 - **Power supply noise**
- **These can be reduced by making the sensor larger**
 - **This increases power dissipation**
 - **But we may want many sensors**

“Reasonable” Values

- **Based on conversations with engineers at Sun, Intel, and HP (Alpha)**
- **Linearity: not a problem for range of temperatures of interest**
- **Slew rate: $< 1 \mu\text{s}$**
 - This is the time it takes for the physical sensing process (e.g., current) to reach equilibrium
- **Sensor bandwidth: $\ll 1 \text{ MHz}$, probably 100-200 kHz**
 - This is the sampling rate; $100 \text{ kHz} = 10 \mu\text{s}$
 - Limited by slew rate but also A/D
 - Consider digitization using a counter

“Reasonable” Values: Precision

- **Mid 1980s: $< 0.1^\circ$ was possible**
 - **Precision**
 - $\pm 3^\circ$ is very reasonable
 - $\pm 2^\circ$ is reasonable
 - $\pm 1^\circ$ is feasible but expensive
 - $< \pm 1^\circ$ is really hard
 - **The limited precision of the G3 sensor seems to have been a design choice involving the digitization**
- } P: 10s of mW

Calibration

- **Accuracy vs. Precision**
 - Analogous to mean vs. stdev
- **Calibration deals with accuracy**
 - The main issue is to reduce inter-die variations in offset
- **Typically requires *per-part* testing and configuration**
- **Basic idea: measure offset, store it, then subtract this from dynamic measurements**

Dynamic Offset Cancellation

- **Rich area of research**
- **Build circuit to continuously, dynamically detect offset and cancel it**
- **Typically uses an op-amp**

- **Has the advantage that it adapts to changing offsets**
- **Has the disadvantage of more complex circuitry**

Role of Precision

- **Suppose:**
 - Junction temperature is J
 - Max variation in sensor is S , offset is O
 - Thermal emergency is T
- **$T = J - S - O$**

- **Spatial gradients**
 - If sensors cannot be located exactly at hotspots, measured temperature may be G° lower than true hotspot
- **$T = J - S - O - G$**

Rate of change of temperature

- **Our FEM simulations suggest maximum 0.1° in about 25-100 μ s**
- **This is for power density < 1 W/mm² die thickness between 0.2 and 0.7mm, and contemporary packaging**
- **This means slew rate is not an issue**
- **But sampling rate is!**

A Different Approach: Soft Sensors

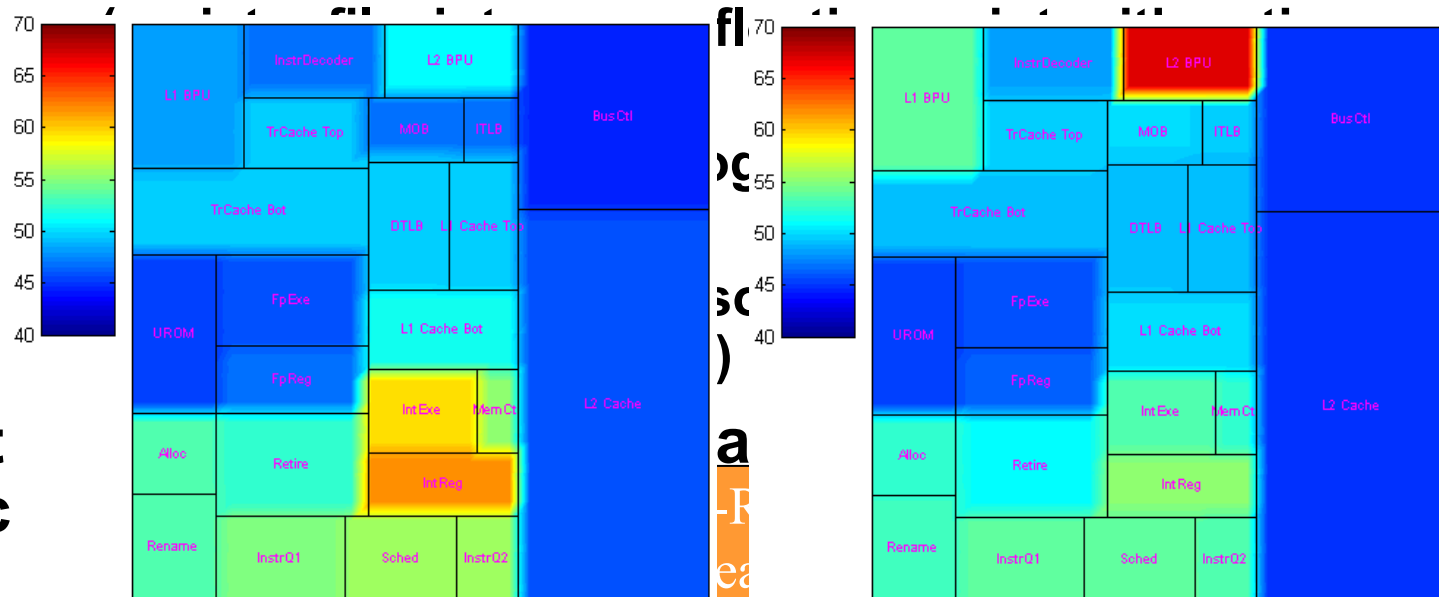
**Supplement “hard” sensor circuits with “soft”
(virtual) sensors using event counts**

Assumes that we know energy cost of events

**Very simple heuristics suffice to estimate
temperature**

CMOS Thermal Sensors

- **DTM requires precise and spatially accurate localized temperature sensing**
 - **Precise: avoid false positives/negatives**
 - Requires sensor proximity
 - **Spatially accurate: hotspots may move according to workload**
 - Different workloads stress different structures



- **But to c**

is really required

Event Counters as Soft Sensors

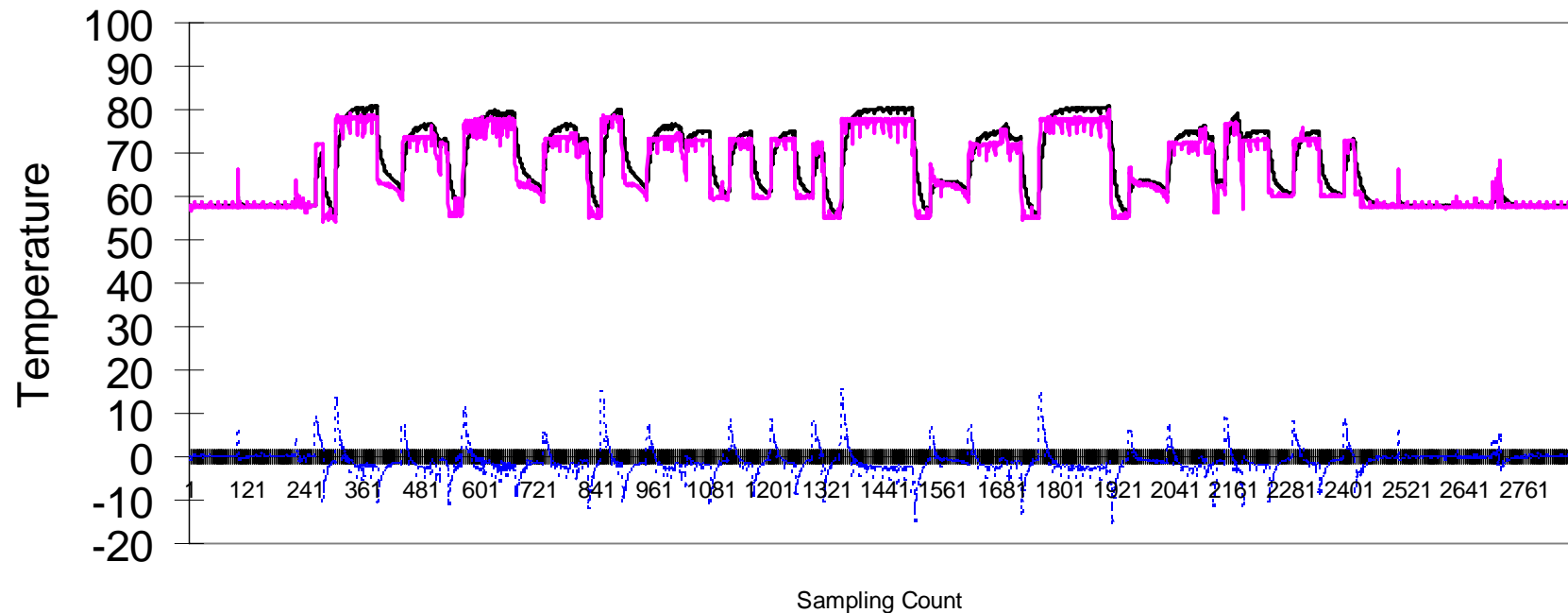
- **Performance counters**
 - Used for profiling and performance tuning
 - Count events like instructions per cycle, cache misses, etc.
 - We know the energy cost of most of these events!
 - We know area of associated structures
 - From this, we can estimate power density and hence change in temperature
- **Simple Regression Analysis**
 - $T = aX + b$
 - The most probable value of Y can be predicted for any value of X
 - Y is temperature
 - X is counter value from the performance counter
 - a and b are constants
 - Computing T is extremely cheap

Related Work

- **Lee and Skadron (ICCD'06)**
 - Validated performance-counter temperature estimation against HotSpot
- **Bellosa (various)**
 - Essentially performs full solution to differential equation
 - Models only a single temperature
- **Han and Koren (TACS'06)**
 - Present an alternative, efficient implementation for using event counters
- **This work shows that very simple linear regression can accurately estimate temperature**
 - Necessary for soft sensors to be viable

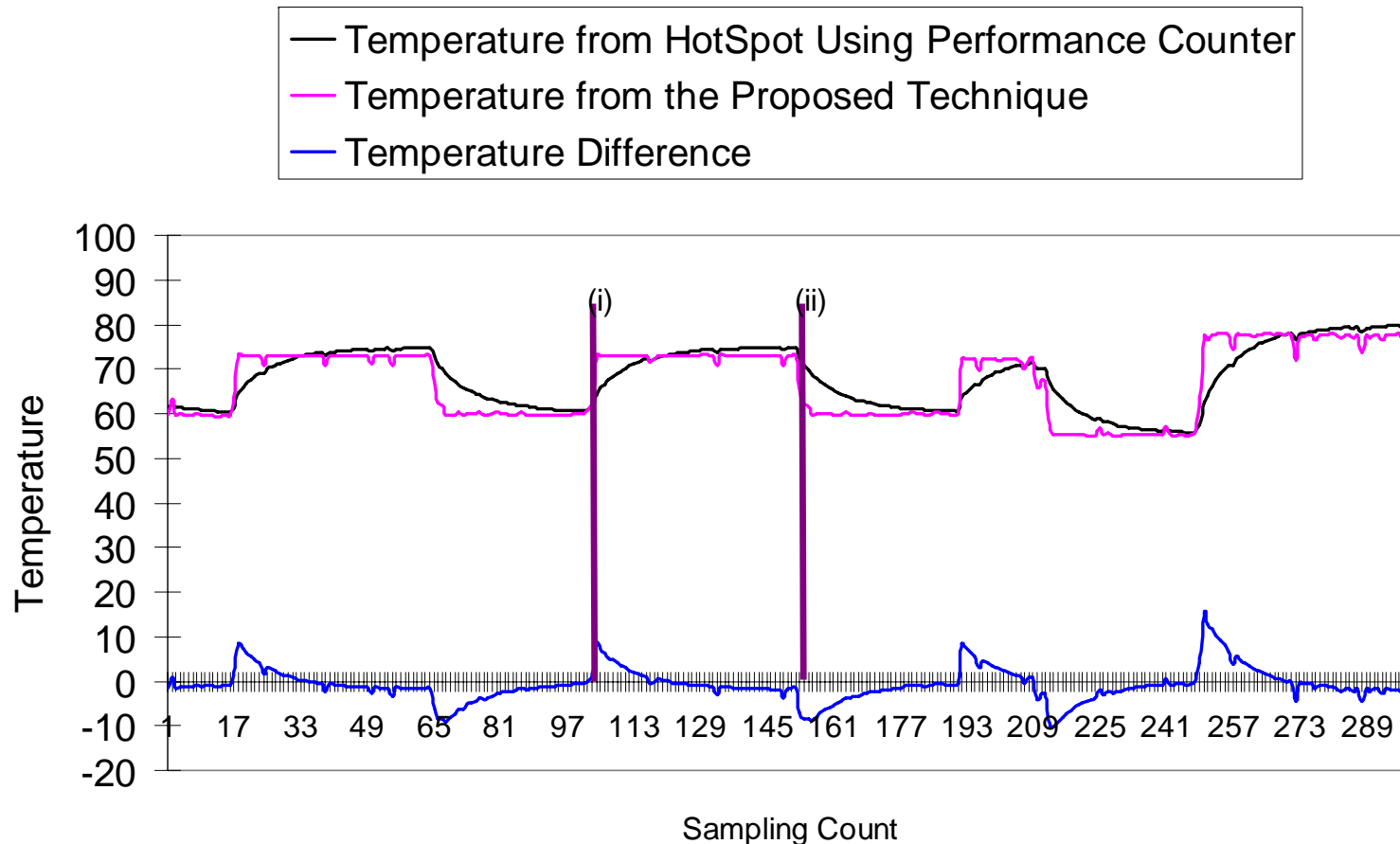
Accuracy Evaluation – bzip2

— Temperature from HotSpot Using Performance Counter
— Temperature from the Proposed Technique
- - - Temperature Difference



- **Close agreement, except on phase boundaries**

Accuracy Evaluation – bzip2



- Linear model overestimates temperature rate of change
- **This could actually be beneficial for DTM** as a way to implement *predictive* response; recent work has suggested this reduces impact of throttling
 - (Srinivasan and Adve, ICS'03)

Conclusions re Soft Sensors

- **Allocating CMOS thermal sensors to all the potential local hotspots may be too costly**
- **But tracking local hotspots is necessary for security and reliability**
- **“Soft” sensors can augment a smaller number of hard sensors**
 - **Based on the event counters like those already embedded in most processors**
 - **Low cost, can monitor multiple sites**
 - **Regression calculation is cheap**
 - **May be especially well suited for predictive throttling and temperature-aware scheduling**
- **ITHERM 2006**

Implications and Issues

- **Can't really use existing performance counters**
 - Interferes with other performance monitoring
 - This work: proof of concept to show value of soft sensors
- **Need targeted, dedicated event counters**
 - Cost of event counter + linear regression vs. CMOS sensor???
- **Soft sensors need calibration too**
 - Use calibrated hard sensor(s) as reference, calibrate on bootup

Sensors Summary

- **Sensor precision cannot be ignored**
 - Reducing operating threshold by 1-2 degrees *will* affect performance
- **Precision of 1° is conceivable but expensive**
 - Maybe reasonable for a single sensor or a few
- **Precision of 2-3° is reasonable even for a moderate number of sensors**
- **Power and area are probably negligible from the architecture standpoint**
- **Sampling period $\leq 10\text{-}20 \mu\text{s}$**
- **“Soft” sensors are promising**

Overall Conclusions

- ***Power-aware and temperature-aware design are different***
- **Temperature-aware design requires a temperature model**
- **HotSpot well suited to pre-RTL modeling**
- **Temperature-aware design needs to**
 - **minimize performance impact**
 - **maximize thermal uniformity**
- **Sensor issues are important**