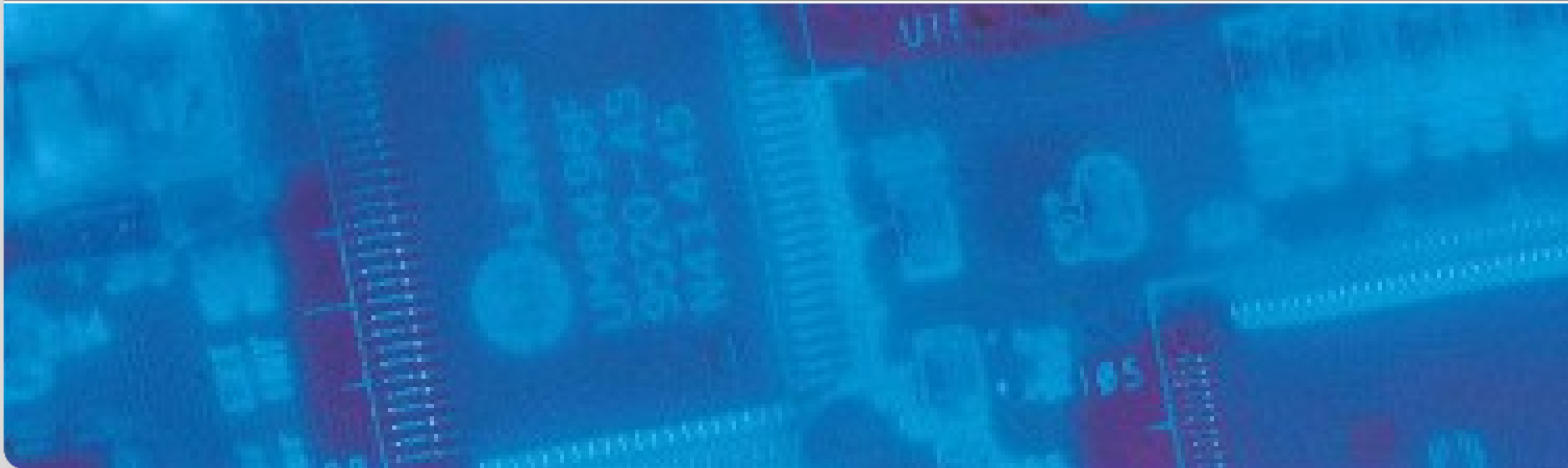


Low Power Design

Volker Wenzel on behalf of Prof. Dr. Jörg Henkel
Summer Term 2016

CES – Chair for Embedded Systems





Overview Low Power Design Lecture

- Introduction and Energy/Power Sources (1)
- Energy/Power Sources(2): Solar Energy Harvesting
- Battery Modeling – Part 1
- Battery Modeling – Part 2
- Hardware power optimization and estimation – Part 1
- Hardware power optimization and estimation – Part 2
- Hardware power optimization and estimation – Part 3
- Low Power Software and Compiler
- Thermal Management – Part 1
- **Thermal Management – Part 2**
- Aging Mechanisms in integrated circuits
- Lab Meeting

- Lab Meeting July, 21th 2016 9:45
 - Technologiefabrik; Haid-Und-Neu-Straße 7; 2nd floor
 - relevant for the oral examination
- Information about the oral examination:
 - <http://ces.itec.kit.edu/972.php>
 - request appointment 4-6 weeks in advance
 - e.g. via email: exam-ces@ira.uka.de
 - the exam need not take place in this semester

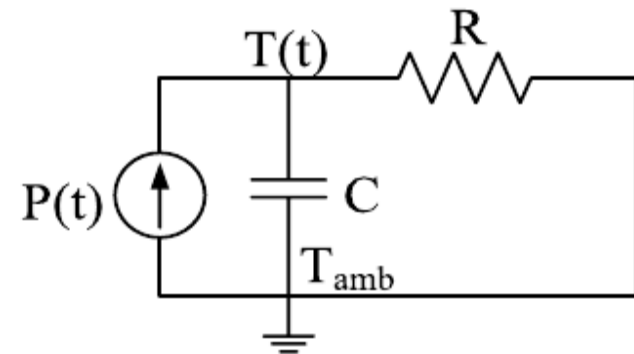
- The RC-model
- Thermal Simulation with HotSpot
- Thermal Sensors
- Thermal Management
- 3D Integration
- Vorlesungsevaluation

Correction: Temperature and Reliability

- Transient errors may result due to timing errors
 - Approx. 5% **increase** in delay every 10°C temperature increase [Xie 2006]
 - Timing errors result from spatial temperature variations
 - localized hotspots need to be avoided
 - Clock trees are particularly vulnerable
 - Span across multiple thermal areas
 - Additional buffers can be inserted to cope with thermal clock skew

(src.: [Chakraborty, 2008])

The RC-Model

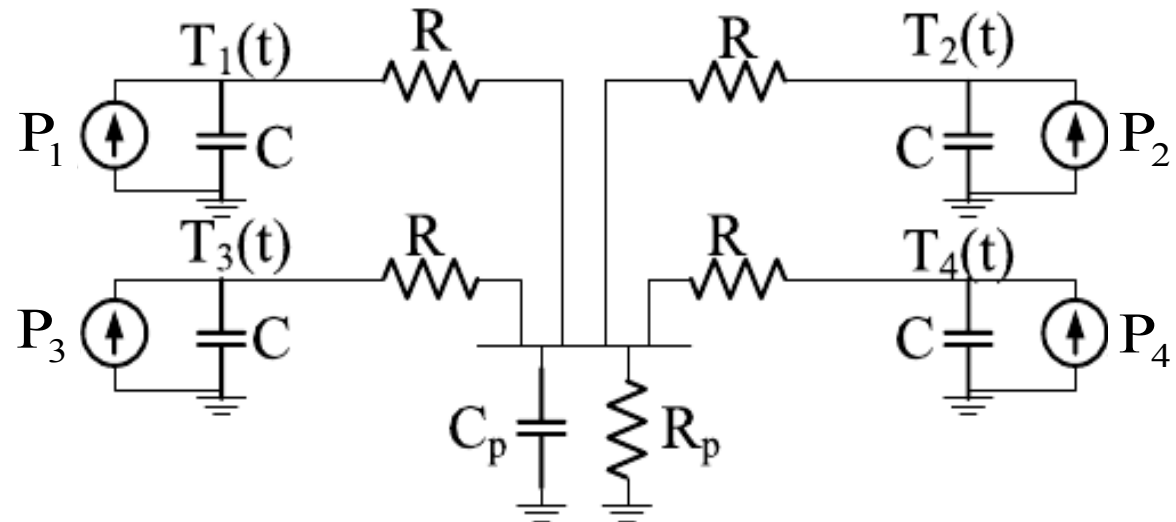


RC equivalent thermal circuit for single component with heat dissipating, e.g. through packaging

Voltage \triangleq Temperature

Current \triangleq Heat dissipation

This gives us the thermal equation from last week as:

$$\frac{dT}{dt} = -\frac{T}{RC} + \frac{P}{C}$$


RC equivalent thermal circuit for four components with heat dissipating to outside through package (C_p , R_p)

(src.: [Shi 2010])

The RC Model (cont)

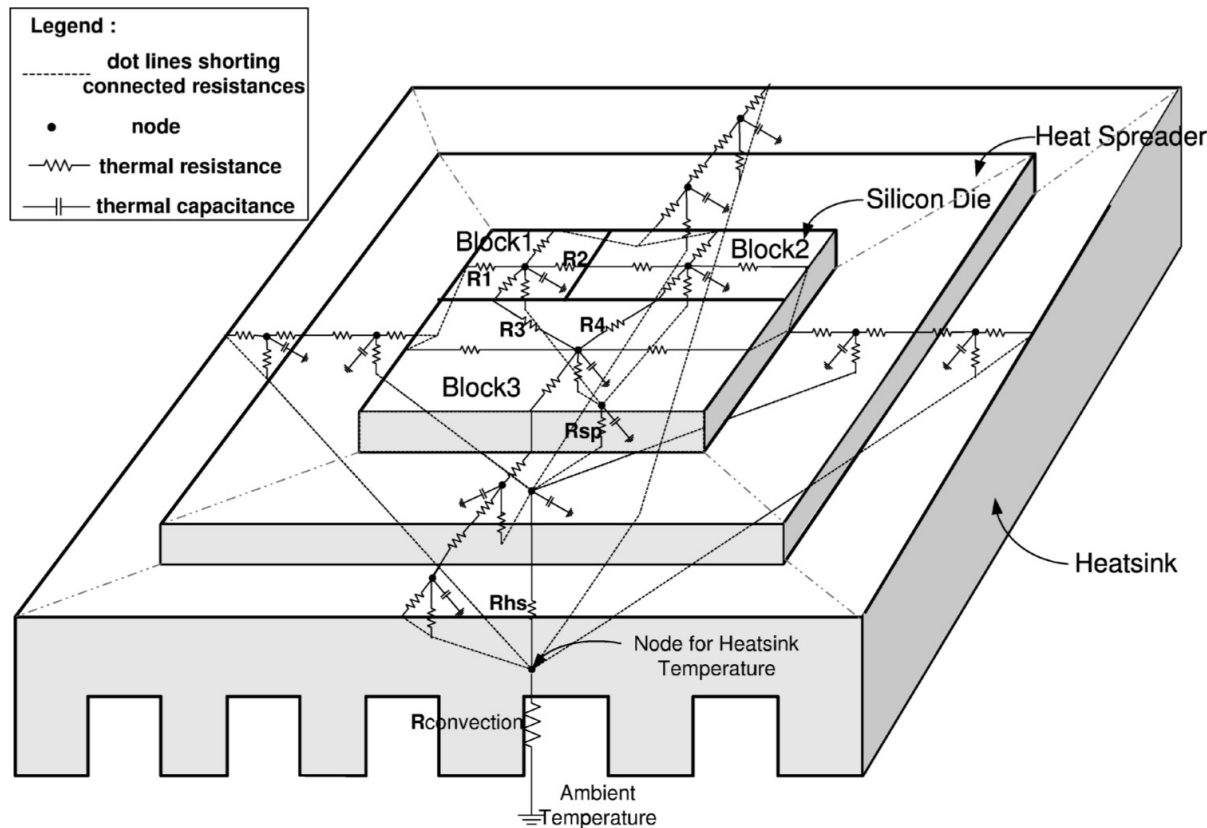


Fig. 1. Example HotSpot RC model for a floorplan with three architectural units, a heat spreader, and a heat sink. The RC model consists of three layers: die, heat spreader, and heat sink. Each layer consists of a vertical RC pair from the center of each block down to the next layer and a lateral RC pair from the center of each block to the center of each edge.

(src.: [Skadron, 2004])

The RC Model (cont)

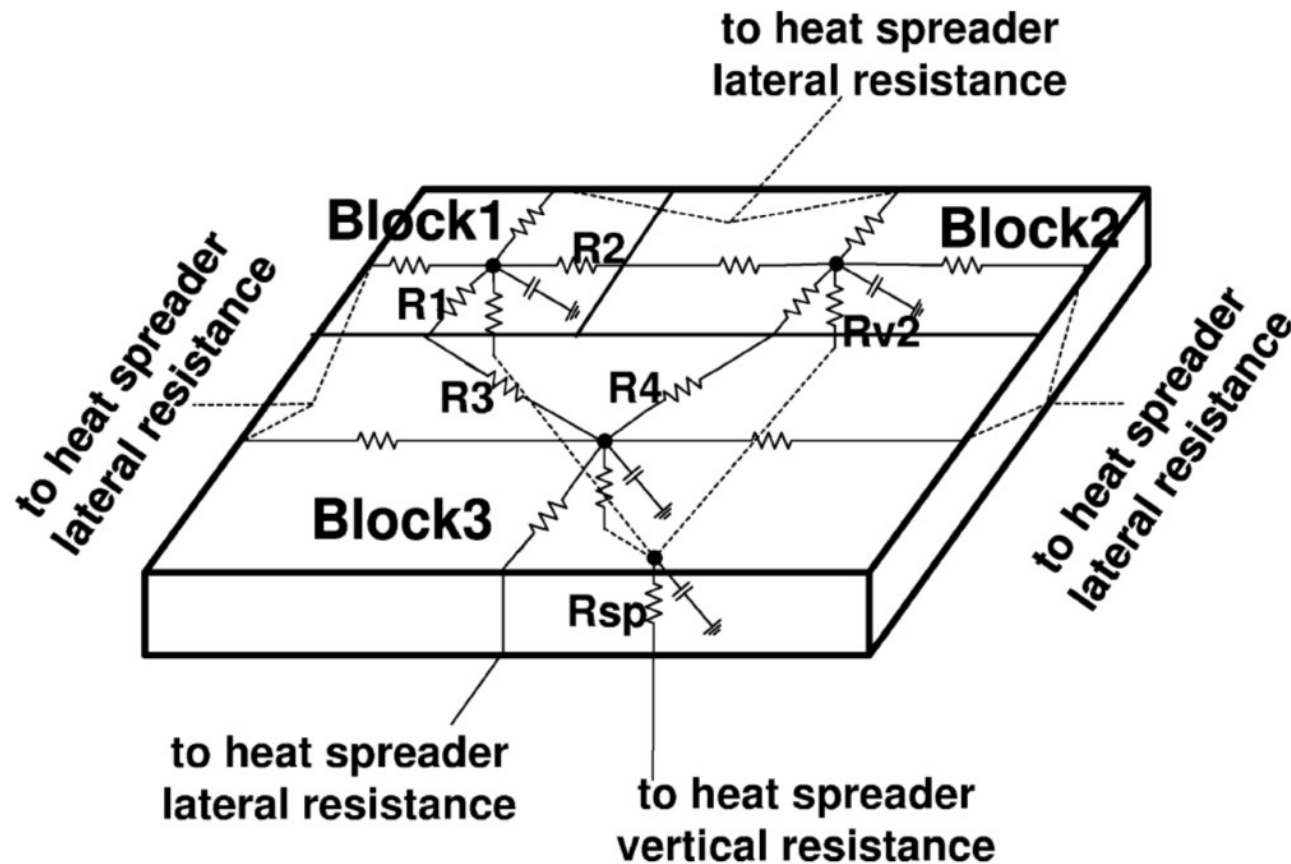


Fig. 2. The RC model for just the die layer.

(src.: [Skadron, 2004])

Thermal Simulation

- Thermal simulators such as **HotSpot** calculate thermal distribution by solving equation of RC equivalent model
- Accuracy of simulation dependent on the granularity of components
 - **Block** based: coarse granularity (CPU, cache, etc.), **fast**
 - **Grid** based: divides blocks into smaller parts, slower, more accurate temperature distribution, **slow**
- Accuracy also dependent on the power input!
 - **Instruction-based** simulators count execution of instructions and know power consumption of each block
 - E.g. Wattch, gem5, McPAT
 - inaccurate but fast (Wattch inaccuracy up to 30%) [Brooks 2000]
 - Circuit-based simulators
 - highly accurate but very slow



HotSpot



[[Home](#) | [Documentation](#) | [Version History](#) | [FAQ](#) | [How-To](#) | [License](#) | [People](#) | [Download](#) | [Links](#)]

Announcing version 6.0!

Version 6.0 introduces several new features that can be useful to special thermal modeling needs: 1) a upgraded solver based on SuperLU that significantly speeds up steady-state simulations; 2) an improved 3D model that supports layers with non-uniform thermal resistivity and heat capacity; 3) an improved secondary heat transfer path model that is compatible with 3D system. You can download version 6.0 [here](#).

What is HotSpot?

HotSpot is an accurate and fast thermal model suitable for use in architectural studies. It is based on an equivalent circuit of thermal resistances and capacitances that correspond to microarchitecture blocks and essential aspects of the thermal package. The model has been validated using finite element simulation. HotSpot has a simple set of interfaces and hence can be integrated with most power-performance simulators like Wattch. The chief advantage of HotSpot is that it is compatible with the kinds of power/performance models used in the computer-architecture community, requiring no detailed design or synthesis description. HotSpot makes it possible to study thermal evolution over long periods of real, full-length applications.

Why thermal modeling?

With power density and hence cooling costs rising exponentially, temperature-aware design has become a necessity. Processor packaging is becoming a major expense, and for many

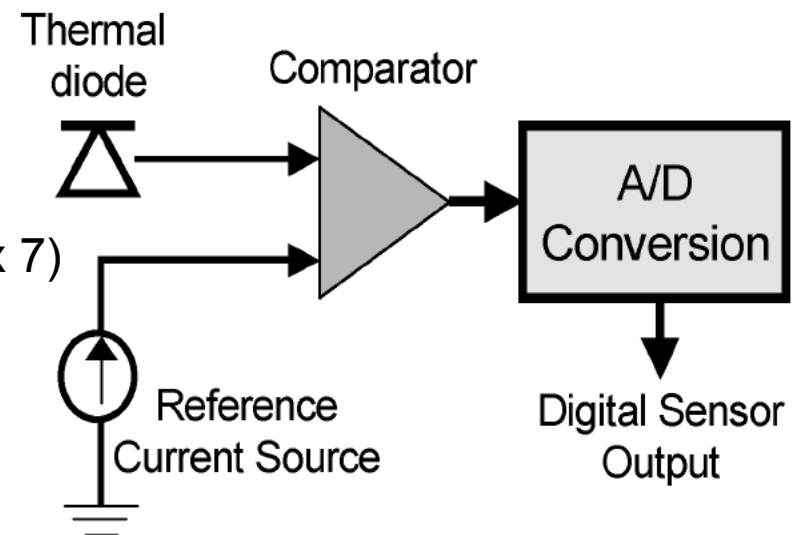
Thermal Sensors: Thermal Diodes

- Currently most common method for on-chip thermal measurement

- Used by Intel, AMD, Xilinx, etc..
- typical accuracy: $\pm 4^{\circ}\text{C}$ (Xilinx Virtex 7)

- Analog circuitry

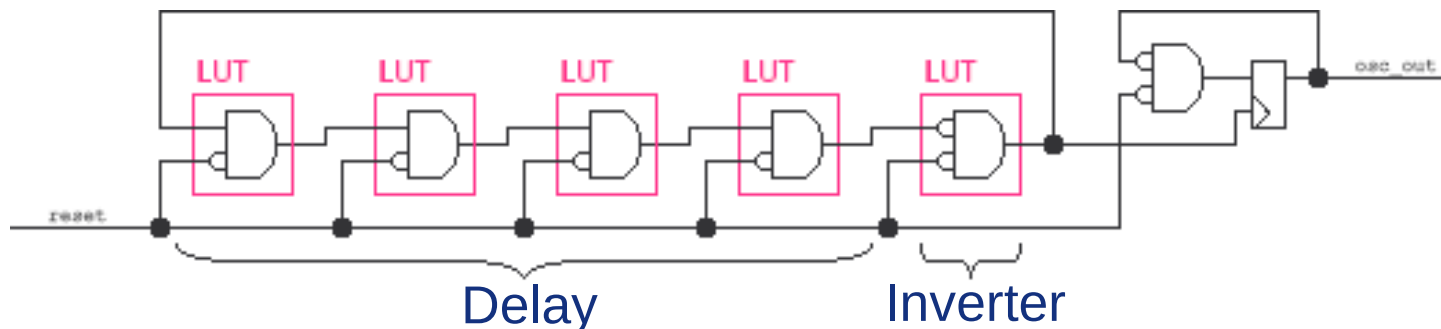
- needs A/D converter
- occupies large chip area



(src.: [Long, 2008])

Thermal Sensors: Ring Oscillator

- Idea: analyze negative thermal side-effects to quantify temperature
- Due to increased delay ring oscillators oscillate slower at higher temperatures
 - Oscillation frequency determined using a reference clock
 - Provide **relative** temperature values
 - Challenge: must be calibrated to obtain absolute values
 - Jitter
- Xilinx reference design:



(src: [Xilinx])

Thermal Sensors: Leakage based

- Leakage is temperature dependent
- Idea: Measure leakage to determine temperature

$$I_{DSUB}(T) = I_{S0}(T) e^{\frac{V_{GS} - V_{TH}(T)}{n k T / q}} \left(1 - e^{\frac{-V_{DS}}{k T / q}} \right)$$

(src.: [Ituero 2008])

Thermal Sensors: Leakage based (cont'd)

Idea: measure the time a capacitor takes to discharge capacitance through leakage current

- Input switches from low-to-high
→ M1 transitions from “on” to “off”
→ Charge stored in CL should remain, but slowly decreases due to leakage current
- When voltage of CL falls below a threshold, the inverter M3-M4 produces a low-to-high transition
- Temperature can be determined by the delay between the input and output transitions

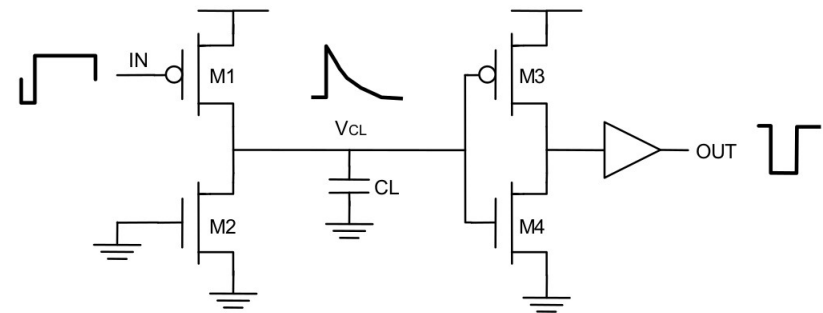


Fig. 1. Subthreshold current thermal sensor.

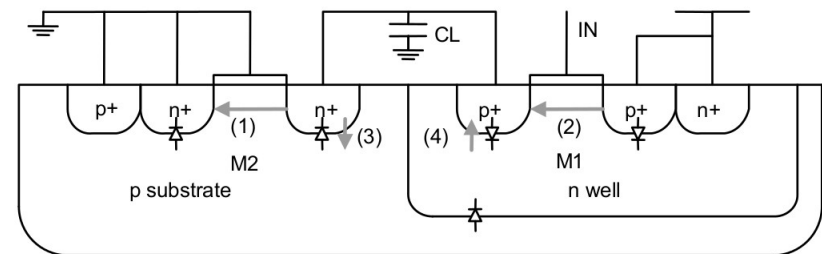


Fig. 2. Leakage current mechanisms in the thermal sensor.

(src.: [Ituero 2008])

Multi-core thermal management

- Classification of thermal management approaches:
 - **Reactive** approaches
 - Depend on the current temperature
 - **Proactive** approaches
 - Predict the temperature
 - Aim to balance temperature to avoid hotspots
- Naïve reactive approaches:
 - [Skadron, ISCA.2004] controls the temperature by:
 - Switching off the hottest core and turning on the coldest one,
 - but that leads to:
 - **Thermal Cycling** and large spatial variations
 - Negative effect on the performance.

Reactive approaches (cont'd)

- [Coskun, 2007] proposed two **OS-level** methods that achieve temperature-aware task scheduling.

- First method: **Cooltest-FLP**

- Depends on the current temperature and floor-plan.

For each
ready job

- Select the coolest processors
- Give priority to processors, whose neighbors are “idle”

- Reduces the hot spots.

- Second method: **probabilistic method**

- Takes into consideration the analysis of the temperature history.

For each
ready job

- Calculates the probability for each core to receive the incoming job

$$P_n = P_{n-1} \pm W$$

Previous probability

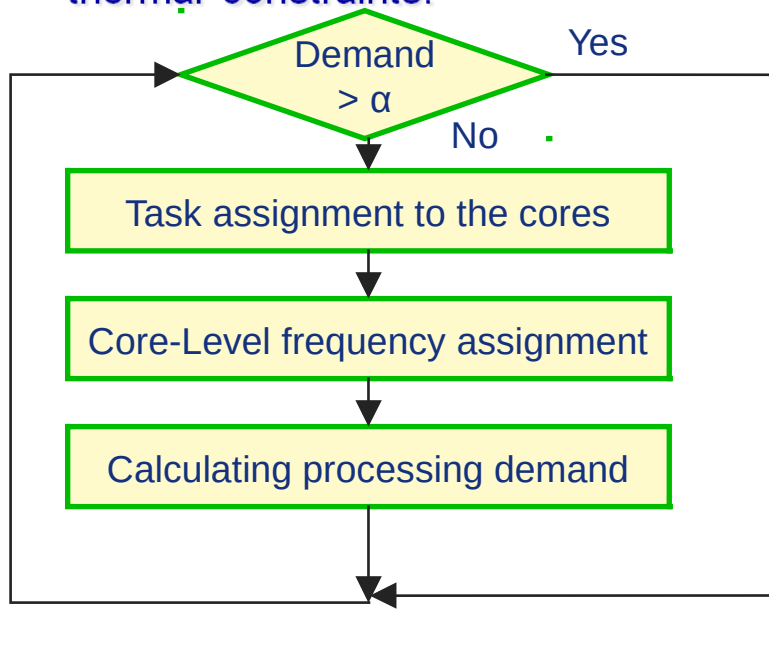
Weight depends on the
core's history

- Achieves more balancing in the temperature and reduces the spatial variation in the temperature

Reactive approaches (cont'd)

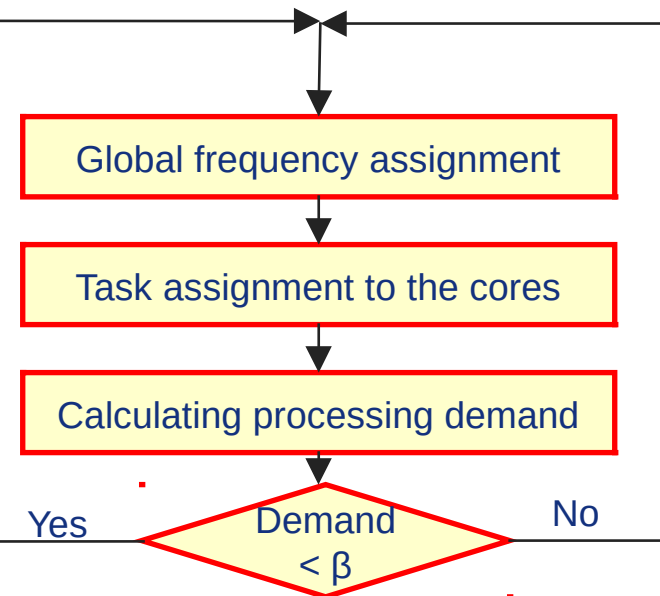
Normal mode:

- ❑ Processing demand < certain threshold.
- ❑ Goal: maximize energy savings with meeting performance demands and thermal constraints.



Thermal balancing mode:

- ❑ Processing demand > certain threshold.
- ❑ Goal: prevent concentration of high power densities, then saving energy.



Reactive approaches (cont'd)

- [Coskun ASPDAC 2008] uses **Integer Linear Programming (ILP)**:
 - Models the applications as tasks graph
 - Results in optimal task scheduling for
 - Given set of tasks with deadlines and dependence constraints
 - Given temperature profiles.
 - Aims at reaching the best temporal and spatial distribution of temperature

Reactive approaches (cont'd)

TABLE I. SUMMARY OF ALL THE ILP OBJECTIVE FUNCTIONS

Label	ILP Objective	Objective Equation
Min-Th&Sp	Minimizing thermal hot spots and gradients	Minimize $H + G$; $H = \max\{Q_p; p = 1 \dots m, \text{ for a system of } m \text{ cores}\}$ where: $Q_p = \sum_{T_i \in T} \{x_{ip} \sum_{v_k} (q_{ik} y_{ik})\}$ $G = \sum_{p, r \in PU, p \neq r} \{n_{pr} \{ \sum_{i, j \in T, i \neq j} x_{ip} x_{jr} [p_{ij} d_{ij} (\tau_i - s_j) + p_{ji} d_{ji} (\tau_j - s_i)] \}\}$
Min-Th	Minimizing & balancing thermal hot spots	Minimize H ; $H = \max\{Q_p; p = 1 \dots m, \text{ for a system of } m \text{ cores}\}$ where: $Q_p = \sum_{T_i \in T} \{x_{ip} \sum_{v_k} (q_{ik} y_{ik})\}$
Bal-En	Balancing energy consumption	Minimize EN_{max} ; $EN_{max} = \max\{EN_p; p = 1 \dots m, \text{ for a system of } m \text{ cores}\}$ where: $EN_p = \sum_{T_i \in T} \{x_{ip} \sum_{v_k} (e_{ik} y_{ik})\}$
Min-En	Minimizing total energy	Minimize EN_{total} ; $EN_{total} = \{ \sum_{T_i \in T} \sum_{v_k} e_{ik} y_{ik} \} + I_{total}; I_{total}$

TABLE II. VARIABLES USED IN THE ILP

x_{ip} :	Set of 1-0 variables s.t. $x_{ip} = 1$ iff T_i is assigned to PU_p
q_{ik} :	Time spent above threshold temperature while running T_i at v_k
t_i :	WCET of T_i considering the voltage setting
s_i :	Execution start time for T_i
τ_i :	Execution finish time for T_i
p_{ij} :	Set of 1-0 variables s.t. $p_{ij} = 1$ iff T_i starts before T_j
n_{pr} :	Set of 1-0 variables s.t. $n_{pr} = 1$ iff p and r are adjacent cores
d_{ij} :	Set of 1-0 variables s.t. $d_{ij} = 1$ iff $\tau_i \geq s_j$
y_{ik} :	Set of 1-0 variables s.t. $y_{ik} = 1$ iff T_i runs at speed v_k
m_{ij} :	Set of 1-0 variables s.t. $m_{ij} = 1$ iff T_j immediately follows T_i
* s.t.: such that	

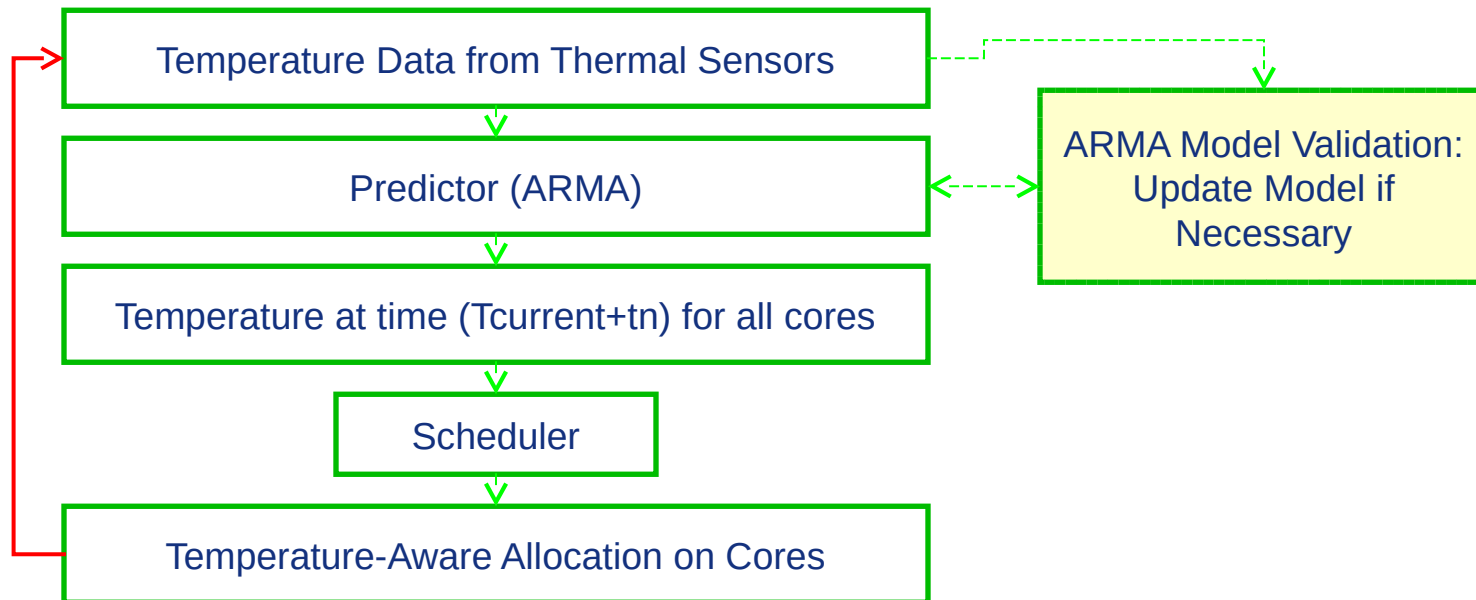
TABLE III. ILP FORMULATION FOR MIN-TH&SP

Minimize $H + G$; $H = \max\{Q_p; p = 1 \dots m, \text{ for a system of } m \text{ cores}\}$ where: $Q_p = \sum_{T_i \in T} \{x_{ip} \sum_{v_k} (y_{ik} q_{ik})\}$ $G = \sum_{p, r \in PU, p \neq r} \{n_{pr} \{ \sum_{i, j \in T, i \neq j} x_{ip} x_{jr} [p_{ij} d_{ij} (\tau_i - s_j) + p_{ji} d_{ji} (\tau_j - s_i)] \}\}$	
Subject to constraints:	
(a) $\forall T_i : \sum_p x_{ip} = 1$	Each task is assigned to only one PU
(b) $\forall T_i : \sum_k y_{ik} = 1$	Each task runs at only one V/f level
(c) $\tau_i = s_i + t_i$	Execution finish time for T_i
(d) $s_i \geq \max_{E_{ji} \in E} \{\tau_j\}$	Task precedence
(e) $\tau_i \leq D_i$	Deadlines for all sink nodes
(f) $s_i \geq \tau_j$; if $p_{ji} = 1$	Precedence for tasks on the same core
(g) $p_{ij} + p_{ji} = 1$; if $x_{ip} = x_{jp} = 1$	If T_i and T_j are scheduled on the same core, either T_i precedes T_j , or vice versa

(src.: [Coskun ASPDAC 2008])

Proactive Approach

- [Coskun 2008] uses autoregressive moving average (ARMA) modeling to:
 - predict the future temperature from history
 - apply thermal-aware job allocation method, which aims to:
 - Avoid reaching a set thermal threshold achieve and balance the temperature across the chip



- ARMA models autocorrelation in a time series

$$y_t + \sum_{i=1}^p (a_i y_{t-i}) = e_t + \sum_{i=1}^q (c_i e_{t-i})$$

y_t - value at time t

e_t - noise/error at time t

a - autoregressive coef.

c - moving avrg. coef.

- Given a stationary stochastic process
→ y_t can be predicted as weighted sum of past values and moving average of error term
- Steps involved:
 - Identification: determine p and q
 - Estimation: determine coefficients a and c
 - Model checking: determine quality of estimated values

- Benefits of ARMA model
 - Model is generated through automated process
→ Does not require in depth thermal knowledge
 - High accuracy achievable with large number of samples (>150)
- Shortcomings
 - Workloads vary over time → temperature is not a stationary function!
 - **Solution:** Thermal sensors are used to check if model is still valid
If not, model is updated at runtime
 - As such: requires thermal sensors on each core

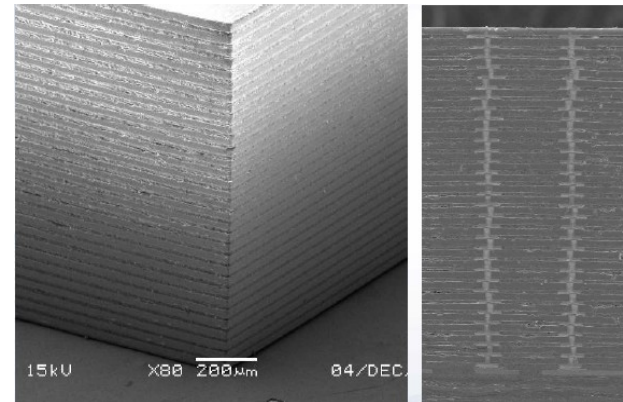
Multicore DTM Strategies & Scalability



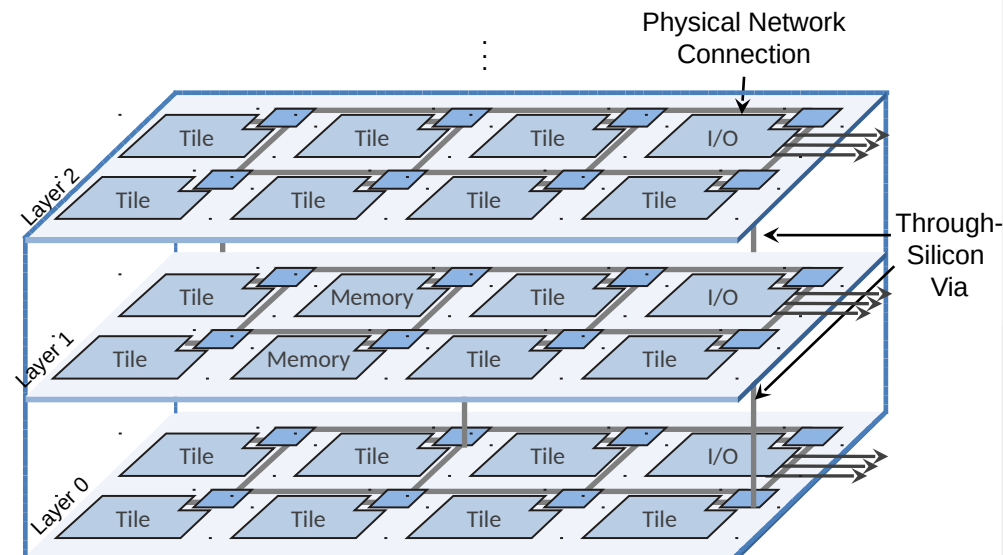
- ❑ **‘Centralized’** management scheme: Manager can use global knowledge but also forms bottleneck for communication as well as computation
→ central point of failure, limited scalability
- ❑ **‘Fully distributed’** scheme: No central bottlenecks. Management is limited by local knowledge
→ can result in local maxima/minima
- ❑ **Hierarchical scheme**: Combines local management with access to global knowledge

3D Architectures

- 3D Integration emerging trend
- Added to the ITRS roadmap in 2009
- Growing research area
- First industry prototypes: IBM, Intel, Xilinx, Samsung...
- Benefits:
 - Decrease in interconnection lengths
 - Higher performance per area



[Source: Samsung]

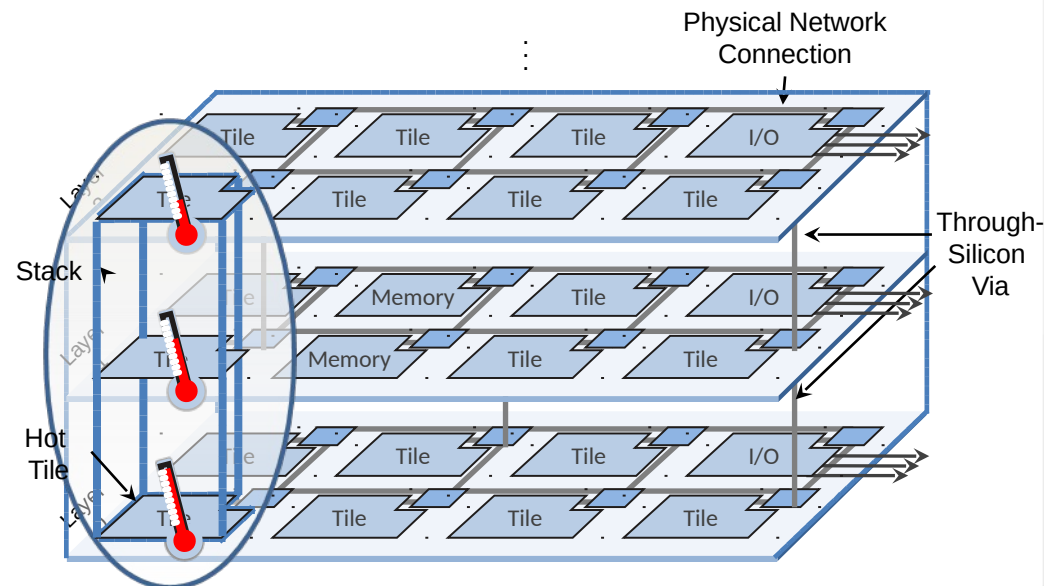


Motivation

- Thermal problems worsens with 3D stacked many-core architectures
- More surface area between cores means more thermal conductivity!
- “Hot” tasks should be running vertically stacked should be avoided
- Methods to increase efficiency of heat dissipation must be examined

Tile consists e.g. of a core, local memory/cache, and interfaces to bus/on-chip network

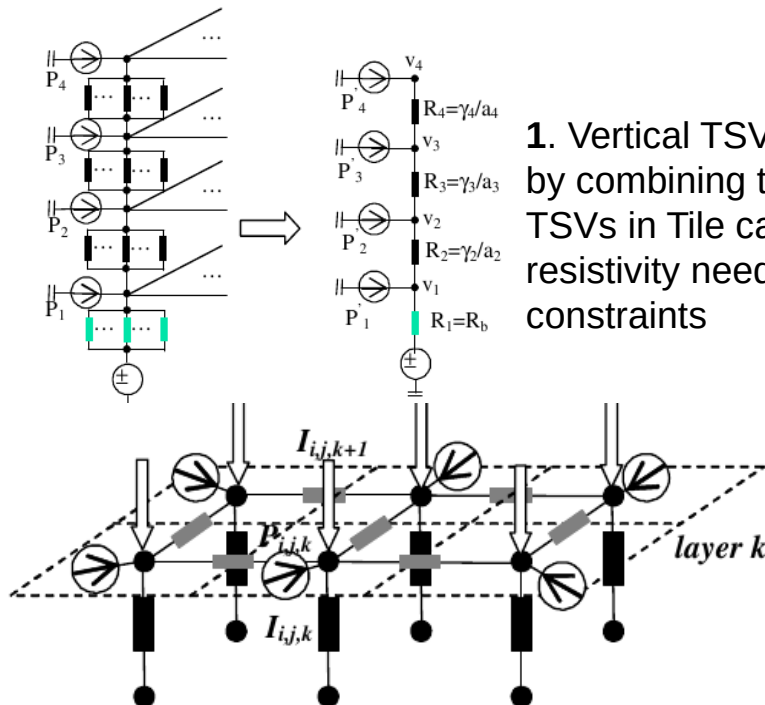
Stack is set of tiles vertically on top of each other



Thermal TSVs

- Through Silicon Vias used as communication links between stacks
- Additional TSVs may be added to increase conductivity to heat sink
 - Etched or drilled through layers
 - Costly to fabricate
 - Occupy large on-chip area (as large as ~20%) with pitch around ~5-10 μ m [Cong 2005]
- TSV planning aims to reduce the number of TSVs while keeping thermal constraints

Alternating direction TSV planning (ADVP) [Cong 2005]



1. Vertical TSV distribution done by combining total resistivities of TSVs in Tile calculating total resistivity needed to keep thermal constraints

2. Horizontal TSV distribution done within each layer to place TSVs near hotspots and maximize heat flow

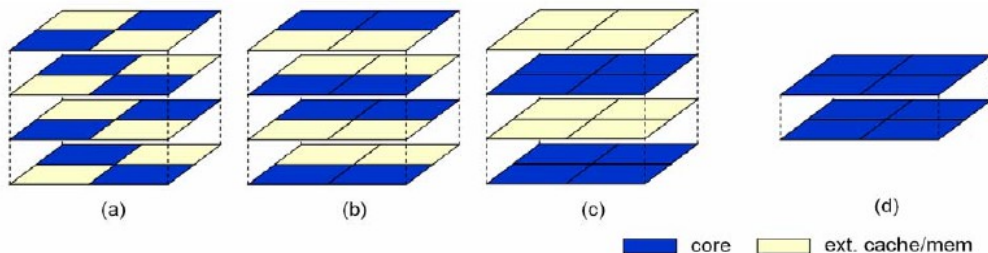
Table 1: ADVP Algorithm

<i>Input:</i> $\Omega = M \times N \times Z$, $I_{i,j,k}$,	
$A_0[k]$ from the previous planning result	
<i>Output:</i> assigned TS-via number for each tile $a_{i,j,k}$	
for every device layer k , starting from the top	
{	
set initial distribution = even distribution of $A_0[k]$, heat propagation for layer k to update $\{I_{i,j,k}\}$	
}	
while not converged	
{	
for each i, j , $1 \leq i \leq M$, $1 \leq j \leq N$,	
1.	vertical TS-via distribution for $\{tile_{i,j,k} 1 \leq k \leq Z\}$
for every device layer k , starting from the top	
{	
2.	horizontal TS-via distribution for $\{tile_{i,j,k} 1 \leq i \leq M, 1 \leq j \leq N\}$ heat propagation for layer k to update $\{I_{i,j,k}\}$
}	
}	

(src.: [Cong, 2005])

Results show up to 68% reduction of TSVs!

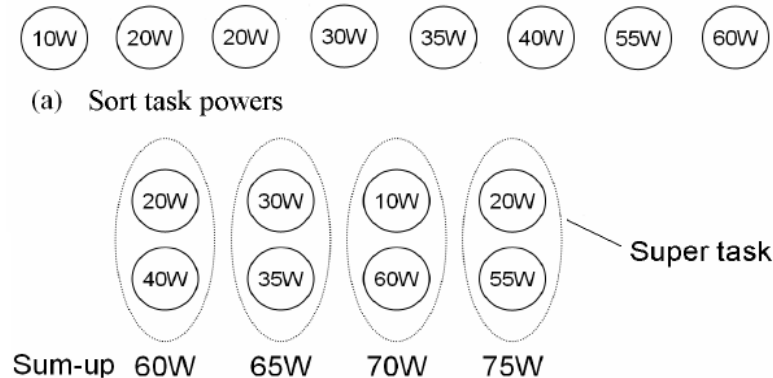
- Floorplanning can play a key role



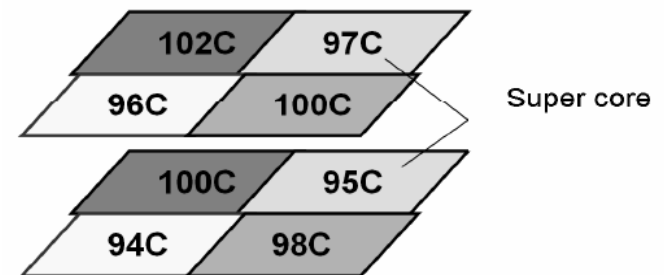
[Zhou 2008]

- Temperature balancing by stack

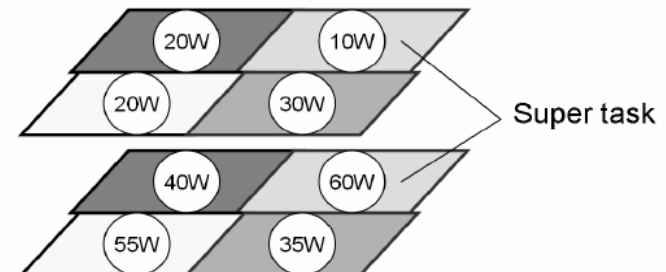
- Results:
max Temperature: 121°C
- Baseline Linux 2.6 scheduler
max Temperature: 145°C
→ reduction of 24°C



(b) Combine tasks into super tasks and sort their powers



(c) Sum up the temperatures of super cores and sort them



(d) Assign super tasks onto super cores

[Zhou 2008]

- ThermOS: 3D multi-core thermal management added to a linux 2.6 kernel
- Based on data acquired through thermal and workload monitoring it applies:

Distributed workload migration (every 20ms):

1. vertically adjacent cores i, k have different cooling efficiency E_i, E_k
if $E_i < E_k$ compare job from job queue of k with min IPC to job in queue of i with max IPC
2. If $\min \text{IPC}(k) < \max \text{IPC}(i)$
Trade tasks between queues
3. Balance jobs between horizontally adjacent cores by comparing average IPCs

Global power-thermal budgeting (every 1-100ms):

Voltages and frequencies are distributed vertically based on the running workloads and thermal impact of cores. Optimal configurations are pre-computed and stored in LUT

In order to ensure thermal constraints are met,
Distributed thermal management makes short-term adjustments using DVFS

- Thermal simulations are often a trade-off between accuracy and simulation time
- Multi-core architectures present new challenges and opportunities for thermal management
→ balancing temperatures can be a very effective technique
- Heat dissipation in 3D Architectures is a major challenge and limits their effectiveness

[Shi 2010] Bing Shi et al, "Dynamic Thermal Management for Single and Multicore Processors Under Soft Thermal Constraints," ISPED 2010.

[Skadron 2004] K. Skadron et al, "Temperature-Aware Microarchitecture: Modeling and Implementation." ACM Transactions on Architecture and Code Optimization, 1(1):94-125, Mar. 2004.

[Brooks 2000] D. Brooks et al, "Wattch: A Framework for Architectural-Level Power Analysis and Optimizations. International Symposium on Computer Architecture, 2000.

[Ituero 2007] P. Ituero et al, "Leakage-based On-Chip Thermal Sensor for CMOS Technology," Circuits and Systems, 2007. ISCAS 2007

[Long 2008] J. Long et al, "Thermal monitoring mechanisms for chip multiprocessors," ACM Trans. Architect. Code Optim., Aug. 2008

[Coskun 2008] A. Coskun et al. "Proactive Temperature Balancing for Low Cost Thermal Management in MPSoCs." ICCAD 2008.

[Coskun ASPDAC 2008] A.. Coskun et al, "Temperature-Aware MPSoC Scheduling for Reducing Hot Spots and Gradients." ASPDAC 2008.

[Coskun 2007] A. Coskun et al, "Temperature Aware Task Scheduling in MPSoCs. " DATE 2007.

[Cong 2005] . Cong and Y. Zhang. "Thermal via planning for 3-d ics." ICCAD 2005.

[Zhou 2008] X. Zhou et al, "Thermal management for 3d processors via task scheduling. In Parallel Processing," ICPP 2008

[Zhu 2008] C. Zhu et al, "Three-dimensional chip-multiprocessor run-time thermal management." Computer-Aided Design of Integrated Circuits and Systems, Aug. 2008.