Update

• Today: Thermal modeling issues, will not cover all slides but included for reference
• Monday (10/11): No Class
• Wed (10/13): Di/Dt Modeling/Design
Thermal issues

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
- Ion, Ioff, Igate, 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!

Resulting performance

25% - 30% extra performance (110°C to 0°C)
Reliability

The Arrhenius Equation: \( MTF = A \times \exp\left(\frac{E_a}{K \times T}\right) \)

- \( MTF \): 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
Heat mechanisms

• Conduction
• Convection
• Radiation
• Phase change
• Heat storage
Conduction

- Similar to electrical conduction (e.g. metals are good conductors)
- Heat flow from high energy to low energy
- Microscopic (vibration, adjacent molecules, electron transport)
- No major displacement of molecules
- Need a material: typically in **solids** (fluids: distance between mol)
- Typical example: thermal “slug”, spreader, heatsink

\[
q_k = \frac{(T_1 - T_2)kA}{L} \quad \theta_k = \frac{L}{kA}
\]
Conduction

Different materials (not a strong function of temperature)
Si – more variation

Convection

- Macroscopic (bulk transport, mix of hot and cold, energy storage)
- Need material (typically in fluids, liquid, gas)
- Natural vs. forced (gas or liquid)
- Typical example: heatsink (fan), liquid cooling

\[ q_c = \overline{h_c} A (T_s - T_\infty) \]

\[ \theta_c = \frac{1}{\overline{h_c} A} \]

Radiation

- Electromagnetic waves (can occur in vacuum)
- Negligible in typical applications
- Sometimes the only mechanism (e.g. in space)

\[
q_r = A_1 \varepsilon_{1,2} \sigma (T_1^4 - T_2^4)
\]
\[
\theta_r = \frac{T_1 - T_2}{A_1 \varepsilon_{1,2} \sigma (T_1^4 - T_2^4)}
\]

Surface-to-surface contacts

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

Phase-change

Thermal solutions evolution:
• Natural air cooling
• Forced-air cooling
• Liquid cooling
• Phase change (e.g. heat pipe)
• Refrigeration

Phase change:

a. Solid changing to a liquid—fusion, or melting,
b. Liquid changing to a vapor—evaporation, also boiling,
c. Vapor changing to a liquid—condensation,
e. Liquid changing to a solid—crystallization, or freezing,
f. Solid changing to a vapor—sublimation,
g. Vapor changing to a solid—deposition.
Thermal capacitance

• Example:

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

\[ C_p \text{(Aluminum)} = 875 \text{ J/(kg-°C)} \]

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

\[ C_{\text{bulk}} = V \cdot C_p \cdot \rho = 59.28 \text{ J/°C} \]
Dynamic Compact Thermal Model

Electrical-thermal duality

\[ V \approx \text{temp (T)} \]
\[ I \approx \text{power (P)} \]
\[ R \approx \text{thermal resistance (Rth)} \]
\[ C \approx \text{thermal capacitance (Cth)} \]
\[ RC \approx \text{time constant} \]

KCL:

differential eq. \[ I = C \cdot \frac{dV}{dt} + \frac{V}{R} \]

difference eq. \[ \Delta V = \frac{I}{C} \cdot \Delta t + \frac{V}{RC} \cdot \Delta t \]

thermal domain \[ \Delta T = \frac{P}{C} \cdot \Delta t + \frac{T}{RC} \cdot \Delta t \]

\[ T = T_{\text{hot}} - T_{\text{amb}} \]

- One can compute stepwise changes in temperature for any granularity at which one can get P, R, C
- RC network \( \Rightarrow \) matrix form of these equations
Example System

Heat sink

Heat spreader

PCB

Die

Interface material

IC Package

Pin
Lateral Model

- Determined by the floorplan and the length of shared edges between adjacent blocks
3D Model (Lateral and Vertical)

Legend:
- dot lines shorting connected resistances
- node
- thermal resistance
- thermal capacitance

Node for Heat Sink Temperature

Heatsink

Silicon Die

Heat Spreader

Interface material (not shown)
Combined package model

**Steady-state**
- **Tj** – junction temperature
- **Tc** – case temperature
- **Ts** – heatsink temperature
- **Ta** – ambient temperature

Validation

- Validated and calibrated using MICRED test chips
  - 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 a lot
Dynamic Thermal Management

• Goal:
  • Provide dynamic techniques to cool chip when needed
  • Exploit natural variations due to different applications, phase behavior, ...
  • Allow designers to target average, rather than worst-case behavior

• Design Decisions:
  • Mechanism & policy for triggering response?
  • What should response be?
  • How to select DTM trigger levels?
Power consumption impacts cost

• System costs associated with power dissipation:
  - Thermal control cost
    – Heatsinks, fans
  - Power delivery
    – Power supply
    – Decoupling caps…

From: Gunther, et al.
“Managing the Impact of Increasing Microprocessor Power Consumption,” Intel Technology Journal, Q1, 2001
Average and Worst Case Power

- System costs are constrained by *worst case* power dissipation
- *Average case* power dissipation can often be much lower
  - Aggressive Clock Gating
  - Applications variations
  - Underutilized resources
    - Not enough ILP
    - Floating point units during integer code execution
- Currently about a 30% difference
- Likely to further diverge…
Dynamic Thermal Management

- Designed for Cooling Capacity w/out DTM
- Designed for Cooling Capacity w/ DTM
- DTM Trigger Level

System Cost Saving

Time

Temperature

DTM Disabled

DTM/Response Engaged
DTM: Definitions

- **Initiation Delay** – OS interrupt/handler
- **Response Delay** – Invocation time (e.g. adjust clock)
- **Policy Delay** – Number of cycles engaged
- **Shutoff Delay** – Disabling time (e.g. re-adjust clock)
DTM: When, How, and What

Trigger Mechanism:
When do we enable DTM techniques?

Initiation Mechanism:
How do we enable technique?

Response Mechanism:
What technique do we enable?
DTM: Trigger Mechanisms

• **Mechanism: How to deduce temperature?**

• **Direct approach:** Temperature sensors providing feedback
  - Implemented in some PowerPC chips (G3, G4) [Sanchez, 1997]
  - Sensor quantity, placement, and precision will be discussed later

• **Other indirect approaches possible**

---

**Policy: When to begin responding?**

- Trigger level set too high: Packaging cost will be high
  - Little advantage
- Trigger level set too low
  - Frequent triggering causes performance to suffer
- Choose trigger level to exploit difference between average and worst-case power.
DTM: Initiation Mechanisms

• Operating system or microarchitectural control?
  • Hardware support can significantly reduce performance penalty

• Policy Delay Settings
  • For Volt/Freq scaling, much of the performance penalty can be attributed to enabling/disabling
  • Increasing policy delay reduces overhead; smarter initiation techniques would help as well
DTM: Response Mechanisms

• Scaling Techniques
  • Clock Frequency Scaling [Intel Pentium 4]
  • Voltage and Frequency Scaling
  • Temperature-tracking frequency scaling [Skadron03]
    – Adjusts frequency to account for T-dep. of switching speed

• Microarchitectural Techniques
  • Speculation Control [Manne98]
  • Low-Power Cache Techniques [Huang00]
    – Hierarchical Responses
  • Decode Throttling [Sanchez97]
  • Fetch Toggling [Brooks01]
  • Feedback controlled Fetch Gating [Skadron02]
  • Migrating Computation [Skadron03]
  • Dual Pipelines [Lim02]
Dynamic Voltage/Frequency Scale

- Voltage Scheduler predicts workload requirements
- Set frequency/voltage to near-optimal, energy savings

Burd, et al., ISSCC2000
- 5MHz @ 1.2V: 6 MIPS, 2.8mW
- 80MHz @ 3.8V: 85 MIPS, 460mW
- 70us 1.2V <-> 3.8V

Transmeta Crusoe
- Commercial implementation (500-700MHz, 1.2-1.6V)
Temperature-Tracking Frequency

Temperature affects:

- Transistor threshold and mobility
- Ion, Ioff, Igate, delay
- ITRS: 85°C for high-performance, 110°C for embedded!

So adjust frequency as \( f(T) \) -- **TTDFS**
Speculation Control

- **Manne et al. (ISCA ’98)**
  - Branch confidence estimator used to determine whether to speculate
  - Pipeline gating based on confidence estimation
  - 38% reduction in wrong-path instructions with ~1% performance loss

- **But Parikh et al. (HPCA ’02) found much smaller savings; ED product is zero or negative**
  - Significant energy savings only come with significant loss of performance
  - This is because many instructions are squashed early in the pipeline, so reduction in wrong-path instructions is not a useful metric
  - Benefit is actually a function of prediction accuracy
    - Only for very badly predicted programs do you get benefit
    - Well-predicted programs suffer
Dynamic Hardware Resizing

- Complexity Adaptive Processors
- Based on application characteristics
  - Underutilized structures may be reduced with minimal performance impact
  - Resize Caches, Issue Queues, etc.
  - Resize => Reduce Capacitance => Reduce Energy
  - Of course, this only helps manage heat if it reduces heat dissipation within hot spots
    - And does so for a sufficiently long duration
DEETM

• Dynamic Energy Efficiency and Temperature Management
  • Slack algorithm detects if slowdown can be tolerated
    • If so, invoke techniques to reduce energy
  • Temperature algorithm
    • If temperature limit is reached, invokes techniques
• Techniques considered
  • Filter Cache, Voltage Scaling, etc.
Control-theoretic DTM

- **Fetch toggling**
  - disable fetch every N cycles
  - 4/5, 2/3, 1/2, 1/3, 1/5, ...

- How to set the fetch rate?
  - (Assume idealized temperature sensing)
Feedback-Control of Fetch Toggling

- Formal feedback control

\[ m = K_C (e + K_I \sum e + K_D \frac{de}{dt}) \]

- easy to compute
- toggling = f(m)

PID:  

Controller: I-fetch toggling

Actuator:

Thermal dynamics

setpoint  \( e \)

measured T

Temp. sensor

P  \( m \)  T
Formal Feedback Control

- Regulatory control problem: hold value to a specified setpoint
  - Example: temperature

  - Proved that PID controller will not allow temperature to exceed setpoint by more than 0.02°
    - Max power dissipation, thermal dynamics, sampling rate $\Rightarrow$ max overshoot
    - This precision is excessive but illustrates the value of formal feedback control theory
Performance Loss

- Performance loss reduced by 65%
Migrating Computation

- When one unit overheats, migrate its functionality to a distant, spare unit (MC)
  - Spare register file (Skadron et al. 2003)
  - Separate core (CMP) (Heo et al. 2003)
  - Microarchitectural clusters
  - etc.

- Raises many interesting issues
  - Cost-benefit tradeoff for that area
  - Use both resources (scheduling)
  - Extra power for long-distance communication
  - Floorplanning
# Migrating Computation – Reg File

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>FPMap</td>
<td>IntMap</td>
<td>IntQ</td>
<td>LdStQ</td>
<td>IntReg</td>
</tr>
<tr>
<td>FPMul</td>
<td>FPQ</td>
<td>ITB</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPReg</td>
<td>DTB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FPAdd</td>
<td>I-Cache</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bpred</td>
<td>D-Cache</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Diagram:

- IntReg2
- LdStQ
- IntMap
- FPMap
- FPQ
- ITB
- DTB
- Icache
- Dcache

- IntQ
- IntReg
Thermal Scheduling (Cai 2002)

- **Primary pipeline:** maximal performance, complex pipeline structure
- **Second pipeline:** Minimum power and energy consumption, very simple in order structure and target mobile anywhere-anytime applications.
- **Transparent to OS and applications**
- **Maximal utilizing on die clock/power gating for energy saving**

Diagram:

- Majority mobile apps with performance requirements
- Text email, caller-id, reminder and other none high performance w/ anywhere-anytime requested apps
Scheduling Algorithm (Cai 2002)

S1: Normal Operation (Primary Pipeline)
S2: Stall Fetch & Clear Pipeline
S3: Alternate Operation (Secondary Pipeline)
S4: Disable Clock or Scale F-V
Hybrid DTM

• DVS is attractive because of its cubic advantage
  • $P \propto V^2f$
  • This factor dominates when DTM must be aggressive
  • But changing DVS setting can be costly
    – Resynchronize PLL
    – Sensitive to sensor noise $\Rightarrow$ spurious changes

• “ILP techniques” are attractive because they can use instruction level parallelism to hide/reduce impact of DTM
  • Only effective when DTM is mild

• So use both!
  • Need to find “crossover point”
Hybrid DTM, cont.

- Combine fetch gating with DVS
  - When DVS is better, use it
  - Otherwise use fetch gating
  - Determined by magnitude of temperature overshoot
  - Crossover at FG duty cycle of 3
  - FG has low overhead: helps reduce cost of sensor noise

![Graph showing slowdown vs. duty cycle for fetch gating (FG) and DVS with a crossover at FG duty cycle of 3.](image-url)
Hybrid DTM, cont.

• DVS doesn’t need more than two settings for thermal control
  • Lower voltage cools chip faster

• FG by itself does need multiple duty cycles and hence requires PI control

• But in a hybrid configuration, FG does not require PI control
  • FG is only used at mild DTM settings
  • Can pick one fixed duty cycle

• This is beneficial because feedback control is vulnerable to noise
Simulation Details

- 85°C maximum temperature
  - Guard band requires a trigger threshold of 81.8°
- Ambient temperature (inside computer case): 45°C
- $R_{\text{package}} = 0.8$ K/W (old package model)
  - 0.7 K/W necessary if DTM not available
- Die thickness: 0.5mm
- Currently neglecting interface material
- 9 SPEC2000 benchmarks, both integer and FP
  - 4 hover near 81.8°C, rest are above
- SimpleScalar/Wattch, modified to model pipeline and power of an Alpha 21364 as closely as possible
- Scaled to 130nm, 1.3V, 3.0 GHz
Performance Comparison

- TT-DFS is best but can’t prevent excess temperature
  - Suitable for use with aggressive clock rates at low temp.
- Hybrid technique reduces DTM cost by 25% vs. DVS (DVS overhead important)
- A substantial portion of MC’s benefit comes from the altered floorplan, which separates hot units
Conclusions so far

- DTM can be used to reduce cooling costs
- Proper modeling is required
  - HotSpot is publicly available at http://lava.cs.virginia.edu/HotSpot
- ILP matters
- Hybrid techniques beneficial
  - Merge advantages of different schemes
  - Simplify control
- Architectural techniques important in thermal design
- Growing use of clusters and redundant units opens an incredibly rich design space
DTM: Summary and Key Issues

• Dynamic optimizations translate max-power problem to average-power problem
• Heightens importance of average-power techniques like clock gating
• Key Issues:
  • Initiation interval
  • Collection of possible response mechanisms
Thermal monitor

- Based on clock throttling
- Full operational mode: maximal frequency
- Minimal operation mode: clocks are stalled for a part of the duty cycle
- Activation options:
  - By OS (e.g., ACPI)
  - By a special hardware
Thermal sensors

- Two thermal sensors
- Maximal temperature reached $\rightarrow$ throttling
- Critical shutdown point reached $\rightarrow$ shutdown