Hardware-Software Codesign

11. Thermal-Aware Design

Iuliana Bacivarov & Lothar Thiele
Contents

- Why is it important to consider temperature in system design?
- Power and temperature models
- Thermal simulation
- Thermal-aware scheduling
**Power/Thermal Wall**

*Power/Thermal wall* is recognized as the most significant barrier towards high performance.
Multi-Cores Face the Power/Thermal Wall Too

72-Core Intel Xeon Phi platform

[Loh: 3D-Stacked Memory Architectures for Multi-Core Processors, 2008]
Some Solutions

- **VLSI design and cooling solutions**
  - Thermal-aware design, materials, reduce leakage and switching, ...
  - Use better heat sinks, fans, air cooling, liquid cooling

- **Thermal management**
  - Voltage/frequency scaling
  - Stop-go execution
    - completely TURN OFF components to allow for cooling
  - Migration of tasks from hot to cool area

[source: Wikipedia]

[MJPEG decoder on 25-core processor]
But scheduling of jobs and thermal management techniques affect both *timing* and *thermal* properties.

Thermal and performance objectives must be considered simultaneously during design.
Some Design Questions

Thermal and performance objectives must be considered simultaneously during design

- How can we simultaneously consider during design both timing and temperature?

- What is the worst case peak temperature of the chip?

- What is an optimal temperature-aware mapping scheme?

- What are temperature-aware scheduling techniques with low overhead (simple control, no temperature sensors)?
Why is it important to consider temperature in system design?

Power and temperature models

Thermal simulation

Thermal-aware scheduling
Single Source Power Model

- Frequently used power model for constant voltage

\[ P(t) = \phi \cdot T(t) + \psi(t) \]

\[ P(t) = \begin{cases} 
P_a(t) = \phi^a \cdot T(t) + \psi^a, & \text{for active processing} \\
P_i(t) = \phi^i \cdot T(t) + \psi^i, & \text{for idle mode} 
\end{cases} \]

Just leakage power

Including both dynamic and leakage power

Silicon chip

Single power source

Active processing

Idle mode

temperature

power
**Power Model**

\[ P(t) = \phi \cdot T(t) + \psi(t) \]

\[ P(t) = \begin{cases} 
  P^a(t) = \phi^a \cdot T(t) + \psi^a, & \text{for active processing} \\
  P^i(t) = \phi^i \cdot T(t) + \psi^i, & \text{for idle mode} 
\end{cases} \]

**Dynamic power**

- Independent of temperature
- Different power consumption for every code segment
- Separate power consumption for each component (core, cache, memory, …)

**Leakage power**

- Independent of the load
- Depends on the current temperature of the component
- Model [Skadron et al. 2004]
  \[ P_{\text{Leak}} \sim T^2 \cdot e^{-C/T} \]
- For the remaining lecture: We use a linear approximation (see above).
Static Power / Dynamic Power Ratio

![Graph showing the static power/dynamic power ratio for different temperature levels and technology nodes (180nm, 130nm, 100nm, 90nm, 80nm, 70nm). The x-axis represents temperature (K), and the y-axis represents percentage. Each technology node has a distinct line color and symbol.](image-url)
Static Power and Dynamic Power

Dynamic power consumption:

\[ P_{\text{dynamic}} = \alpha CV_{DD}^2 f \]

- Total capacity
- Supply voltage
- Clock frequency

Between 0 and 1; quantifies switching activity

Static power consumption:

- 20% or more in sub-micron era
- Mostly leakage, i.e., the power dissipated by a transistor whose gate is intended to be off
Single Power Source Model

\[ \frac{dT}{dt} = \frac{C}{P - G(T - T_{amb})} \]

\[ P(t) = \phi \cdot T(t) + \psi(t) \]

\[ \frac{dT}{dt} = -gT + h \quad \text{with} \quad g = \frac{G - \phi}{C}, \quad h = \frac{\psi + GT_{amb}}{C} \]

Silicon chip

Cooling

Single power source

Thermal conductance

Thermal capacity

Environment temperature

Power parameters
Solution of the Thermal Equation

Explicit solution

\[ T(t) = T^\infty + (T(t_0) - T^\infty) \cdot e^{-\frac{G - \phi}{C} \cdot (t - t_0)} \]

\[ T^\infty = \frac{GT_{amb} + \psi}{G - \phi} \]

Steady state temperature
Temperature Profile

**Active state**: dynamic and static (leakage) power

**Temperature increase**: based on linear thermal model

\[
C \frac{dT}{dt} = -G(T - T_{amb}) + P \\
T(t) = T^\infty + (T(t_0) - T^\infty) \cdot e^{-\frac{G-\phi}{C} \cdot (t-t_0)}
\]
Temperature Profile

Idle state: static (leakage) power

Temperature decrease: based on linear thermal model

\[
C \frac{dT}{dt} = -G(T - T_{amb}) + P
\]

\[
T(t) = T^{\infty} + (T(t_0) - T^{\infty}) \cdot e^{-\frac{G - \phi}{C} \cdot (t - t_0)}
\]
Temperature Profile

Task execution schedule

Peak temperature

Temperature (K)

Time (s)
Multi Source Models

- A and B are matrixes
- T is an N-dimensional temperature vector
- u is the input vector

\[ \frac{dT(t)}{dt} = A \cdot T(t) + B \cdot u(t) \]
Multi Source Models – Solution

Explicit solution:

\[ T(t) = e^{A \cdot t} \cdot T^0 + \int_{-\infty}^{\infty} H(t - \xi) \cdot u(\xi) \, d\xi \]

- \( A, H, B \), are matrixes

- \( T \) is an N-dimensional temperature vector

- \( H_{ij}(t) \) is the impulse response between power injected at source \( j \) and temperature variation at location \( i \)

Impulse response matrix

\[ H(t) = e^{A \cdot t} \cdot B \]
Multi-Core Effect

\[ T_k(t) = T_k^{\text{init}}(t) + \sum_{\ell=1}^{n} T_k,\ell(t) \]

**Neighboring effect #1**

**Self-heating effect**

**Neighboring effect #2**

**Neighboring effect #3**
The Impulse Response

Temperature rises with power at same location (without delay)

Temperature rises with power at some other location after delay
Multi-Core Effect – Heat Transfer (I)

\[ C \cdot \frac{dT(t)}{dt} = (P(t) + K \cdot T^{amb}) - (G + K) \cdot T(t) \]

\[ P(t) = \phi \cdot T(t) + \psi(t) \]

\[ P_\ell(t) = \begin{cases} P^a_\ell(t) = \phi_\ell \cdot T_\ell + \psi^a_\ell & \text{if } S_\ell(t) = 1, \\ P^i_\ell(t) = \phi_\ell \cdot T_\ell + \psi^i_\ell & \text{if } S_\ell(t) = 0. \end{cases} \]

\[ \frac{dT(t)}{dt} = A \cdot T(t) + B \cdot u(t) \]

\[ \begin{align*} A &= -C^{-1} \cdot (G + K - \phi) \\ B &= C^{-1} \quad \text{or} \quad C^{-1} \\ u(t) &= \psi(t) + K \cdot T^{amb} \end{align*} \]

\[ u(t) = \text{input vector} \]
Multi-Core Effect – Heat Transfer (II)

\[ T(t) = e^{A \cdot t} \cdot T^0 + \int_{-\infty}^{\infty} H(t - \xi) \cdot u(\xi) \, d\xi \]

\[ H(t) = e^{A \cdot t} \cdot B \]

closed-form solution of the temperature

\[
\begin{align*}
H_{kk}(t) &= e'_k \cdot e^{A \cdot t} \cdot e_k \cdot B_{kk} = \text{self-impulse response} \\
H_{k\ell}(t) &= \text{impulse response between nodes } \ell \text{ and } k
\end{align*}
\]
Multi-Core Effect – Heat Transfer (III)

\[ T(t) = e^{A \cdot t} \cdot T^0 + \int_{-\infty}^{\infty} H(t - \xi) \cdot u(\xi) \, d\xi \]

Closed-form solution of the temperature

\[ H(t) = e^{A \cdot t} \cdot B \]

\[ T_{\text{init}}(t) = e^{A \cdot t} \cdot T^0 \]

\[ T_k(t) = T_k^{\text{init}}(t) + \sum_{\ell=1}^{n} T_{k,\ell}(t) \]

Temperature of node \( k \)

\[ T_{k,\ell}(t) = \int_{0}^{t} H_{k\ell}(t - \xi) \cdot u_{\ell}(\xi) \, d\xi \]

Convolution between the impulse response \( H_{k\ell} \) and the input \( u_{\ell} \)

\[ u_{\ell}(t) = S_{\ell}(t) \cdot u_{\ell}^a + (1 - S_{\ell}(t)) \cdot u_{\ell}^i \]

Workload of component \( \ell \)

\[
\begin{align*}
    u_{\ell}^a &= \psi_{\ell}^a + K_{\ell\ell} \cdot T^{\text{amb}} \\
    u_{\ell}^i &= \psi_{\ell}^i + K_{\ell\ell} \cdot T^{\text{amb}}
\end{align*}
\]

\[ S_{\ell}(t) = \begin{cases} 
    1 & \text{component } \ell \text{ is ‘active’}, \\
    0 & \text{component } \ell \text{ is ‘idle’}.
\end{cases} \]
Multi-Core Effect

\[ T_k(t) = T_k^{\text{init}}(t) + \sum_{\ell=1}^{n} T_{k,\ell}(t) \]

\[
\begin{align*}
\mathbf{T}^{\text{init}}(t) &= e^{A \cdot t} \cdot \mathbf{T}^0 \\
T_{k,\ell}(t) &= \int_{0}^{t} H_{k\ell}(t - \xi) \cdot u_{\ell}(\xi) \, d\xi
\end{align*}
\]
Solving the Differential Equations

What’s happening in numerical simulations?

\[ \frac{dT(t)}{dt} = A \cdot T(t) + B \cdot u(t) \]

\[ \frac{\Delta T(t)}{\Delta t} = A \cdot T(t_{k-1}) + B \cdot u(t_{k-1}) \]

\[ \Delta T(t) = T(t_k) - T(t_{k-1}) \quad T(t_0) = T_{amb} \]

\[ T(t_k) = T(t_{k-1}) + [A \cdot T(t_{k-1}) + B \cdot u(t_{k-1})] \cdot \Delta t \]

Simulators

- 3D-Ice [http://esl.epfl.ch/3d-ice.html](http://esl.epfl.ch/3d-ice.html)
Contents

- Why is it important to consider temperature in system design?
- Power and temperature models
- Thermal simulation
- Thermal-aware scheduling
Thermal Simulation Tool-Chain

- Application
- Power / Performance Simulator
- Temperature Simulator

Power models of HW components

Modeling of physical structure

- Low-level power/performance simulation/emulation
  - Software: [Benini’05], [Brooks’00]
  - Hardware: [Atienza’07]

- Temperature simulation
  - HotSpot: [Huang’06]
  - 3DICE: [Sridhar’10]

- There are other possibilities as well, e.g. model identification and reduction
How do we consider computation, communication, and memory? How do we link power consumption, time, and temperature? How do we consider scheduling?
procedure
    FIRE(Process p)
    read(INPUT, size, buf)
    manipulate
    write(OUTPUT, size, buf)
end procedure
procedure
   FIRE(Process p)

   read(INPUT, size, buf)
   manipulate
   write(OUTPUT, size, buf)

end procedure
procedure FIRE(Process p)
    read(INPUT, size, buf)
    manipulate
    write(OUTPUT, size, buf)
end procedure
procedure FIRE(Process p)
read(INPUT, size, buf)
manipulate
write(OUTPUT, size, buf)
end procedure

Power Consumption

- ARM Processor 0
- ARM 7 Core
- Memory
- Scratchpad
- Cache
- Bus
- Shared Memory
Thermal Evaluation

### Power Model

<table>
<thead>
<tr>
<th>Time</th>
<th>Tile 1</th>
<th>Tile 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>5ms</td>
<td>$s_{1,p2}$</td>
<td>$s_{1,p1}$</td>
</tr>
<tr>
<td>10ms</td>
<td>Idle</td>
<td></td>
</tr>
<tr>
<td>15ms</td>
<td>$s_{2,p2}$</td>
<td>$s_{1,p3}$</td>
</tr>
<tr>
<td>20ms</td>
<td>26mW</td>
<td>29mW</td>
</tr>
<tr>
<td>25ms</td>
<td>32mW</td>
<td>23mW</td>
</tr>
</tbody>
</table>

### Thermal Model

\[
C \cdot \frac{dT(t)}{dt} = (P(t) + K \cdot T_{amb}) - (G + K) \cdot T(t)
\]

- $P(t) = P = \text{const}, \ 0 \leq t \leq \Delta t$
- $\Delta t = \text{const}$
- Temperature of interest: $T(\Delta t)$

\[
T[k+1] = E \cdot T[k] + F \cdot P[k]
\]

Calculate $E$, $F$ once at the beginning
## Model Data

<table>
<thead>
<tr>
<th>Entity</th>
<th>Parameter</th>
<th>[Unit]</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Power consumption</td>
<td>[W]</td>
<td>Low-level sim.</td>
</tr>
<tr>
<td>Communication queue</td>
<td>Token size</td>
<td>[bytes / access]</td>
<td>Functional sim.</td>
</tr>
<tr>
<td>Processing unit</td>
<td>Clock frequency</td>
<td>[cycles / sec]</td>
<td>Hardware datasheet</td>
</tr>
<tr>
<td></td>
<td>Conductivity matrix</td>
<td>[W/K]</td>
<td>Low-level phy. sim.</td>
</tr>
</tbody>
</table>
Calibration Tool Chain

Sample Mappings → Software Synthesis → Low-Level Power/Timing Simulator → Thermal Architecture Analysis

Execution trace Timing characterization

Power characterization

Thermal platform model: conductivity matrix capacitance matrix

Timing Parameters

Thermal Parameters
(High-Level) Abstract Thermal Simulation

Application

Scheduling Overhead

Idle Task? Store? Restore?

P

C

Ready

Read / Write Occurred

Dispatch

Active

Block-ed

Read / Write Occurred

FIFO Queue Full/Empty
Data for High-Level (Abstract) Thermal Simulation

```
int fire () {
    float i;
    float j;
    read (PORT_IN, &i);
    j = i*i;
    j += 2;
    write (PORT_OUT, &j);
    printf("Wrote: %f\n", j);
    return 0;
}
```

Thermal / Timing Parameters

- **Computing:** $\Delta t = 20\text{ms}, \Delta P = 25\text{mW}$
- **Reading:** $\Delta t = 45\text{ms}, \Delta P = 32\text{mW}$
- **Computing:** $\Delta t = 80\text{ms}, \Delta P = 38\text{mW}$
- **Writing:** $\Delta t = 60\text{ms}, \Delta P = 34\text{mW}$
- **Computing:** $\Delta t = 120\text{ms}, \Delta P = 41\text{mW}$

Analyze hundreds of design alternatives very quickly!
Thermal Simulation Tool Chains

Application

Power / Performance Simulator

Temperature Simulator

Power models of HW components

Modeling of physical structure

Application, Architecture, Mapping

Black-Box

Power models of HW components

Modeling of physical structure
Contents

- Why is it important to consider temperature in system design?
- Power and temperature models
- Thermal simulation
- Thermal-aware scheduling
Multiple Power States

- Power states: trade-off between power consumption and performance

1. Mobile consumer devices
   - Preferred plans:
     - Balanced: Change plan settings
     - Power saver: Change plan settings
     - High performance: Change plan settings
   - Energy savings: Balanced: ⦿⦿⦿⦿⦿, Performance: ⦿⦿⦿⦿⦿
   - Energy savings: Power saver: ⦿⦿⦿⦿⦿, Performance: ⦿⦿⦿⦿⦿
   - Energy savings: High performance: ⦿⦿⦿⦿⦿, Performance: ⦿⦿⦿⦿⦿

2. Server-grade hardware

Source: Windows 7 power management

How to reduce chip temperature without sacrificing performance/timing requirements?
Thermal Control Loop

Reactive Speed Scaling (RSS)

- Speed of processor feedback controlled based on temperature
- Higher temperature $\Rightarrow$ lower speed
- Joint analysis of temperature and timing complex (but possible)
- Temperature sensor can be replaced or extended by a load sensor
Summary

- **Timely behavior** is important in embedded systems, such as avionics, automotive, or media processing
  - Tasks must finish execution within specified deadlines

- **Thermal wall** is recognized as significant barrier to high performance
  - High chip temperatures lead to reliability issues, even higher power consumption, and lower performance.

- **System-level design solutions**
  - Use Dynamic Thermal Management (DTM) to reduce chip temperatures, examples: speed scaling, stop-go scheduling, mapping and migration
  - But without sacrificing timing requirements