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

Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)
Power/Thermal Wall Impede High Performance

Moore’s Law: more transistors/mm²

⇒ More activity/area ⇒ Higher speed \( \checkmark \)
⇒ More power density ⇒ Higher temperature ⇒
⇒ Lower speed \(!!\)

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

48-Core Intel SCC 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
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)?
Contents

► 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 \textit{leakage power} \quad Including both \textit{dynamic and leakage power}
Power Model

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

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

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

### Dynamic power

- Independent on the load
- Depends on the current temperature of the component

### Leakage power

- Model [Skadron et al. 2004]
  \[ P_{\text{leak}} \sim T^2 \cdot e^{-C/T} \]
- Practically, we use a linear approximation [Liu et al. 2007; Chantem et al. 2008]
Static Power / Dynamic Power Ratio

![Graph showing the static power to dynamic power ratio at different temperatures for various technology nodes (180nm, 130nm, 100nm, 90nm, 80nm, 70nm).]
Static Power and Dynamic Power

Dynamic power consumption:

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

- Depends on the wire lengths
- Supply voltage
- Clock frequency
- Between 0 and 1 how often wires transition

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

\[ P = V \left( ke^{-qV_{th}/(ak_{T})} \right) \]
Single Power Source Model

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

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

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

- Thermal conductance
- Thermal capacity
- Environment temperature
Solution of the Thermal Equation (I)

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

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

Idle state: static (leakage) power

Temperature decrease: based on linear thermal model

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

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

Task execution schedule

Peak temperature

Temperature (K)

Time (s)
Solution of the Thermal Equation (II)

- 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

- Explicit solution based on a linear system view

\[ \frac{dT}{dt} = AT + Bu \]

\[ A = -C^{-1}(G - \phi) \]

\[ Bu = C^{-1}(\psi + GT_{amb}) \]

\[ T(t) = T(0)e^{At} + \int_0^t h(t - \tau) Bu(\tau) d\tau \]

\[ h(t) = e^{At} \]

Impulse response
Multi Source Models

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

- **A** and **B** are matrixes
- **T** is an N-dimensional temperature vector
- **u** is the input vector
Multi Source Models – Solution

Explicit solution:

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

Impulse response matrix

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

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 \( i \)

\[ \Delta T_j(t) \]

\[ P_i(t) \]
Multi-Core Effect

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

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

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

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

Power dissipated by component \( L \) in ‘active’ (a) and ‘idle’ (i) processing modes

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

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

u(t) = input vector

C = thermal capacitance matrix
G = thermal conductance matrix
K = thermal ground conductance matrix
P = power dissipation vector
T^{amb} = ambient temperature vector
T^{amb} = T^{amb} \cdot [1, \ldots, 1]'
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

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

Closed-form solution of the temperature:

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

Temperature of node \( k \):

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

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

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

Workload of component \( \ell \):

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

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

Stopping function:

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

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

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

What’s happening in numerical simulations?

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

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

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

With \( P(t) = P = \text{const.} \) for \( 0 \leq t \leq \Delta t \), (and therefore \( u(t) = \text{const.} \))

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

- **Power / Performance Simulator**
  - Low-level power/performance simulation/emulation
    - Software: [Benini’05], [Brooks’00]
    - Hardware: [Atienza’07]
  - Temperature simulation
    - HotSpot: [Huang’06]
    - 3DICE: [Sridhar’10]

- **Temperature Simulator**
  - There are other possibilities as well, e.g. model identification and reduction

- **Application**

- **Power models of HW components**

- **Modeling of physical structure**

![Graph showing temperature over time](image-url)
**High-Level Power Simulation**

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

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
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>s₁,p₂</td>
<td>s₁,p₁</td>
</tr>
<tr>
<td>10ms</td>
<td>Idle</td>
<td></td>
</tr>
<tr>
<td>15ms</td>
<td></td>
<td>s₁,p₃</td>
</tr>
<tr>
<td>20ms</td>
<td>s₂,p₂</td>
<td></td>
</tr>
<tr>
<td>25ms</td>
<td>s₂,p₁</td>
<td></td>
</tr>
</tbody>
</table>

Thermal Model

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

Power Model

\[ P(t) = P = \text{const, } 0 \leq t \leq \Delta t \]
\[ \Delta t = \text{const} \]

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 data-sheet</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
(High-Level) Abstract Thermal Simulation

Application

Scheduling Overhead

Idle Task? Store? Restore?

P

C

P

C

Ready

Active

Blocked

Read / Write Occurred

Dispatch

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

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

Time [s]

Temperature [K]

Application, Architecture, Mapping → Black-Box

Power models

Application

Mapping

Temperature [K]

Time [s]
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
2. Server-grade hardware

Select a power plan

Power plans can help you maximize your computer's performance or conserve energy. Make a plan active by selecting it, or choose a plan and customize it by changing its power settings. Tell me more about power plans

Preferred plans
- Balanced
  - Change plan settings
  - Energy savings: ★★★★
  - Performance: ★★★★★
- Power saver
  - Change plan settings
  - Energy savings: ★★★★★★
  - Performance: ★★★★★
- High performance
  - Change plan settings
  - Energy savings: ★★★★★
  - Performance: ★★★★★★

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
- Guarantees temperature constraints, independent of timing model
Dynamic thermal management (DTM) (e.g., speed scaling, stop-go scheduling, mapping&migration) is a solution to reduce chip temperatures.

- It has to be done without sacrificing timing requirements.

Proper timing and thermal analysis is needed!

Some references:
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
- **Thermal and real-time objectives must be considered simultaneously during design**
  - Efficient timing analysis motivates the use of RSS-like DTM techniques for cool real-time systems