Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs

K. Siozios, D. Rodopoulos and D. Soudris
\{ksiop, drodo, dsoudris\}@microlab.ntua.gr

School of Electrical and Computer Engineering, National Technical University of Athens, Greece
Presentation outline

- Introduction about the importance of thermal analysis
- Limitation of existing approach (Hotspot tool)
  - Applied solely at design time
  - Consequence: don’t take into account constraints posed during application execution that lead to temperature variations
- Proposed methodology for supporting thermal monitoring under run-time constraints
  - Goal: alleviate computational complexity without accuracy degradation
- Experimental results for application mapping onto Altera Stratix-based FPGAs
Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Why temperature is critical?

- Power dissipation has peaked
- This becomes even more important with device scaling
  - Higher power densities

![Diagram showing power density over years with various processors and power densities marked.]
Challenges

- Thermal challenges get worse with:
  - technology scaling, advanced process technologies, higher operation frequencies, etc.

- Design packaging for the worst-case scenario cannot be though as a cost efficient solution

- Thermal modeling requires fine-grain simulation
  - Both in spatial and temporal domain
Current state

- Temperature values can be extracted by:
  - Employing thermal sensors
  - Employing power sensors

- Both of them have advantages & disadvantages
Current state

- Temperature values can be extracted by:
  - Employing thermal sensors
  - Employing power sensors

- Both of them have advantages & disadvantages

- **Power sensors** is the preferable solution for upcoming MPSoC architectures
Current state

- **Hotspot** tool is the state-of-the-art tool for thermal modeling
  - It is based on an equivalent circuit of thermal resistances and capacitances that correspond to microarchitecture blocks and essential aspects of the thermal package.
  - Power traces can be retrieved with power sensors

- **Limitation:**
  - Mentionable runtime and memory footprint overheads !!!
Proposed methodology

- The proposed thermal analysis and exploration methodology seeks to compute:
  - fast and accurate on-chip temperature values

- It consists of two steps:
  - Coarse-grain thermal analysis
  - Fine-grain thermal analysis

- It can be applied either as a monitoring agent at an embedded operating system (OS), or as a stand-alone tool for fast thermal analysis.
Proposed methodology

<table>
<thead>
<tr>
<th>Input</th>
<th>Coarse-grain thermal analysis (Executed once)</th>
<th>Fine-grain thermal analysis (Executed repeatedly)</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>![Application Icon]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Platform</td>
<td>![Platform Icon]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Coarse-grain thermal analysis: We are primarily interested in retrieving a rough and fast estimation of the temperature variation across the chip’s area.
Proposed methodology

**Coarse-grain thermal analysis:** We are primarily interested in retrieving a rough and fast estimation of the temperature variation across the chip’s area.

**Quick_HotSpot:** A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Proposed methodology

**Fine-grain thermal analysis:** We perform detailed thermal analysis only to device regions with increased importance
Proposed methodology

Fine-grain thermal analysis: We perform detailed thermal analysis only to device regions with increased importance
**Proposed methodology**

**Fine-grain thermal analysis**: We perform detailed thermal analysis only to device regions with increased importance

**Quick_Hotspot**: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Proposed methodology

Fine-grain thermal analysis: We perform detailed thermal analysis only to device regions with increased importance
Quick_Hotspot Tool

- Thermal modeling relies on an equivalent thermal RC network which represents the transient and steady state thermal behavior
  - Similar to Hotspot tool
Quick_Hotspot Tool

- Thermal modeling relies on an equivalent thermal RC network which represents the transient and steady state thermal behavior
  - Similar to Hotspot tool

- Thermal analysis can be performed:
  - By combining regions with different detail of accuracy
  - With reduced computational complexity
  - With almost negligible error
Quick_Hotspot Tool

- Thermal modeling relies on an equivalent thermal RC network which represents the transient and steady state thermal behavior
  - Similar to Hotspot tool

- Thermal analysis can be performed:
  - By combining regions with different detail of accuracy
  - With reduced computational complexity
  - With almost negligible error

- The computational complexity of Quick_Hotspot is reduced:
  - each iteration requires $4\times$ less the original number of operations
  - The complexity of rk4_core in the original version of Hotspot is $O(n^2)$.
  - The proposed approach exhibits complexity equals to $O(n)$. 
Source code optimization

- Based on our study we found that we need to focus on the transient solver (the \textit{rk4\_core} function)
  - it is the computationally dominant function at the \textit{Hotspot} tool
  - it performs an iteration of the 4th order \textit{Runge-Kutta} method with adaptive step sizing

\[
\begin{align*}
\bar{k}_1 &= f(P_n, T_n) \Rightarrow \bar{t}_1 = T_n + \frac{h}{2} \times \bar{k}_1 \\
\bar{k}_2 &= f(P_n, \bar{t}_1) \Rightarrow \bar{t}_2 = T_n + \frac{h}{2} \times \bar{k}_2 \\
\bar{k}_3 &= f(P_n, \bar{t}_2) \Rightarrow \bar{t}_3 = T_n + h \times \bar{k}_3 \\
\bar{k}_4 &= f(P_n, \bar{t}_3)
\end{align*}
\]

\[
T_{n+1} = T_n + \frac{h}{6} \times (\bar{k}_1 + 2 \times \bar{k}_2 + 2 \times \bar{k}_3 + \bar{k}_4)
\]
Source code optimization

- Based on our study we found that we need to focus on the transient solver (the \textit{rk4\_core} function)
  - it is the computationally dominant function at the \textit{Hotspot} tool
  - it performs an iteration of the 4th order \textit{Runge-Kutta} method with adaptive step sizing

\begin{verbatim}
rk4\_core(Pn,Tn,k1,h) {
  t1 = Tn + h/2.0 * k1;
  k2 = evaluate_slope(Pn,t1);
  t2 = Tn + h/2.0 * k2;
  k3 = evaluate_slope(Pn,t2);
  t3 = Tn + h * k3;
  k4 = evaluate_slope(Pn,t3);
  Tnew = Tn + h*(k1 + 2*k2 + 2*k3 + k4)/6.0;
  return Tnew;
}
\end{verbatim}

\begin{verbatim}
intended\_approach(Pn,Tn,k1,h) {
  Tnew = Tn + (h * k1);
  return Tnew;
}
\end{verbatim}
Quick_Hotspot Tool

- The complexity of four functionalities were profiled:
  - **Parsing and Initial Configuration**
    - Processing of the input commands and combination of the latter with the configuration file.
  - **R Model Population**
    - Construction of an array with the vertical and lateral thermal resistances between the functional blocks. As we will depict, this step is very important and time-consuming.
  - **Names, Temperature and Power Arrays Initialization**
    - Memory allocation and preparation of power and temperature arrays.
  - **Numerical Part**
    - Numerical iterations.
## Experimental setup

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Slices</th>
<th>Power sources</th>
<th>Array of slices in target FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td>alu4</td>
<td>1,600</td>
<td>1,600</td>
<td>40×40</td>
</tr>
<tr>
<td>apex2</td>
<td>2,025</td>
<td>2,025</td>
<td>45×45</td>
</tr>
<tr>
<td>apex4</td>
<td>1,369</td>
<td>1,369</td>
<td>37×37</td>
</tr>
<tr>
<td>diffeq</td>
<td>1,600</td>
<td>1,600</td>
<td>40×40</td>
</tr>
<tr>
<td>ex5p</td>
<td>3,025</td>
<td>3,025</td>
<td>55×55</td>
</tr>
<tr>
<td>misex3</td>
<td>3,721</td>
<td>3,721</td>
<td>61×61</td>
</tr>
<tr>
<td>s298</td>
<td>2,025</td>
<td>2,025</td>
<td>45×45</td>
</tr>
<tr>
<td>seq</td>
<td>1,849</td>
<td>1,849</td>
<td>43×43</td>
</tr>
<tr>
<td>tseng</td>
<td>1,156</td>
<td>1,156</td>
<td>34×34</td>
</tr>
<tr>
<td><strong>Average:</strong></td>
<td><strong>2,041</strong></td>
<td><strong>2,041</strong></td>
<td><strong>44×44</strong></td>
</tr>
</tbody>
</table>

The target device is an Altera Stratix-based FPGA.

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoc Designs
Number of execution cycles ($\times10^9$) for different tasks of thermal analysis

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Initial configuration</th>
<th>R model population</th>
<th>Initialization</th>
<th>Numerical part</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hotspot</td>
<td>Quick Hotspot</td>
<td>Gain (%)</td>
<td>Hotspot</td>
</tr>
<tr>
<td>alu4</td>
<td>1.02</td>
<td>0.54</td>
<td>47.2%</td>
<td>6,751</td>
</tr>
<tr>
<td>apex2</td>
<td>1.43</td>
<td>1.09</td>
<td>23.6%</td>
<td>13,471</td>
</tr>
<tr>
<td>apex4</td>
<td>0.44</td>
<td>0.41</td>
<td>7.3%</td>
<td>4,180</td>
</tr>
<tr>
<td>diffeq</td>
<td>0.51</td>
<td>0.52</td>
<td>-3.4%</td>
<td>6,651</td>
</tr>
<tr>
<td>ex5p</td>
<td>0.33</td>
<td>0.348</td>
<td>4.8%</td>
<td>2,510</td>
</tr>
<tr>
<td>misex3</td>
<td>0.48</td>
<td>0.46</td>
<td>3.9%</td>
<td>5,771</td>
</tr>
<tr>
<td>s298</td>
<td>1.24</td>
<td>1.56</td>
<td>25.5%</td>
<td>13,493</td>
</tr>
<tr>
<td>seq</td>
<td>1.12</td>
<td>1.04</td>
<td>6.8%</td>
<td>10,257</td>
</tr>
<tr>
<td>tseng</td>
<td>0.26</td>
<td>0.34</td>
<td>29.9%</td>
<td>2,508</td>
</tr>
<tr>
<td>Average</td>
<td>0.76</td>
<td>0.70</td>
<td>7.6%</td>
<td>7,288</td>
</tr>
</tbody>
</table>
Number of execution cycles ($\times 10^9$) for different tasks of thermal analysis

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Initial configuration</th>
<th>R model population</th>
<th>Initialization</th>
<th>Numerical part</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hotspot</td>
<td>Quick Hotspot</td>
<td>Gain (%)</td>
<td>Hotspot</td>
</tr>
<tr>
<td>alu4</td>
<td>1.02</td>
<td>0.54</td>
<td>47.2%</td>
<td>6,751</td>
</tr>
<tr>
<td>apex2</td>
<td>1.43</td>
<td>1.09</td>
<td>23.6%</td>
<td>13,471</td>
</tr>
<tr>
<td>apex4</td>
<td>0.44</td>
<td>0.41</td>
<td>7.3%</td>
<td>4,180</td>
</tr>
<tr>
<td>diffeq</td>
<td>0.51</td>
<td>0.52</td>
<td>-3.4%</td>
<td>6,651</td>
</tr>
<tr>
<td>ex5p</td>
<td>0.33</td>
<td>0.348</td>
<td>4.8%</td>
<td>2,510</td>
</tr>
<tr>
<td>misex3</td>
<td>0.48</td>
<td>0.46</td>
<td>3.9%</td>
<td>5,771</td>
</tr>
<tr>
<td>s298</td>
<td>1.24</td>
<td>1.56</td>
<td>25.5%</td>
<td>13,493</td>
</tr>
<tr>
<td>seq</td>
<td>1.12</td>
<td>1.04</td>
<td>6.8%</td>
<td>10,257</td>
</tr>
<tr>
<td>tseng</td>
<td>0.26</td>
<td>0.34</td>
<td>29.9%</td>
<td>2,508</td>
</tr>
<tr>
<td>Average:</td>
<td>0.76</td>
<td>0.70</td>
<td>7.6%</td>
<td>7,288</td>
</tr>
</tbody>
</table>

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Complexity of employed benchmarks

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Total power sources</th>
<th>Number of cycles for performing thermal analysis ($\times 10^9$)</th>
<th>Gain (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Hotspot</td>
<td>Quick Hotspot</td>
</tr>
<tr>
<td>alu4</td>
<td>1,600</td>
<td>24,466</td>
<td>7,759</td>
</tr>
<tr>
<td>apex2</td>
<td>2,025</td>
<td>23,215</td>
<td>15,876</td>
</tr>
<tr>
<td>apex4</td>
<td>1,369</td>
<td>8,121</td>
<td>4,934</td>
</tr>
<tr>
<td>diffeq</td>
<td>1,600</td>
<td>12,028</td>
<td>7,688</td>
</tr>
<tr>
<td>ex5p</td>
<td>3,025</td>
<td>5,331</td>
<td>3,059</td>
</tr>
<tr>
<td>misex3</td>
<td>3,721</td>
<td>10,644</td>
<td>6,655</td>
</tr>
<tr>
<td>s298</td>
<td>2,025</td>
<td>22,660</td>
<td>15,701</td>
</tr>
<tr>
<td>seq</td>
<td>1,849</td>
<td>17,541</td>
<td>11,726</td>
</tr>
<tr>
<td>tseng</td>
<td>1,156</td>
<td>5,328</td>
<td>3,060</td>
</tr>
<tr>
<td><strong>Average:</strong></td>
<td><strong>2,041</strong></td>
<td><strong>14,371</strong></td>
<td><strong>8,495</strong></td>
</tr>
</tbody>
</table>

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Locate regions of importance

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
Circle denote regions of the device with temperatures values greater than $\frac{T_{\text{max}}}{2}$

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
All the three scenarios result to similar error.
The optimal among them is the modification of area coverage.
When error in temp. is 8%, this reduces the execution cycles about 50%.
Improvement in execution time

- Selective thermal analysis results an additional improvement in the number of cycles at our proposed solution about **43%**, on average.
  - This additional reduction is translated as **75%** reduction in comparison to the original Hotspot tool (for entire architecture).

Figure evaluates the execution cycles with localized thermal analysis
### Accuracy of thermal analysis (1/3)

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Slices with temperatures higher than $\frac{T_{\text{max}}}{2}$</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total</td>
<td>Hotspot</td>
</tr>
<tr>
<td>alu4</td>
<td>1,600</td>
<td>510</td>
</tr>
<tr>
<td>apex2</td>
<td>2,025</td>
<td>1,276</td>
</tr>
<tr>
<td>apex4</td>
<td>1,369</td>
<td>684</td>
</tr>
<tr>
<td>differq</td>
<td>1,600</td>
<td>831</td>
</tr>
<tr>
<td>ex5p</td>
<td>3,025</td>
<td>1,241</td>
</tr>
<tr>
<td>misex3</td>
<td>3,721</td>
<td>1,340</td>
</tr>
<tr>
<td>s298</td>
<td>2,025</td>
<td>1,177</td>
</tr>
<tr>
<td>seq</td>
<td>1,849</td>
<td>1,035</td>
</tr>
<tr>
<td>tseng</td>
<td>1,156</td>
<td>437</td>
</tr>
<tr>
<td><strong>Average:</strong></td>
<td><strong>2,041</strong></td>
<td><strong>948</strong></td>
</tr>
</tbody>
</table>

Quick_Hotspot: A Software Supported Methodology for Supporting Run-Time Thermal Analysis at MPSoC Designs
On average 7 slices (about 0.3% of the fabricated slices) are identified that operate under $\frac{T_{\text{max}}}{2}$ temperature with only one of the two approaches.
Accuracy of thermal analysis (3/3)

- Maximum temperature difference for the same slice with the two alternative thermal analysis tools.
- The accuracy derived by our proposed solution is almost identical to the one achieved by HotSpot 5.0
  - The average maximum error in slices among the benchmarks is only 0.003°C
Conclusions

- A systematic methodology for performing run-time thermal analysis for MPSoC designs was presented.

- The methodology is supported by a new software tool, named *Quick_Hotspot*.

- The proposed methodology was applied to evaluate thermal analysis of alternative floor-plans.

- Experimental results show that our solution achieves significant improvement in computational complexity:
  - ranging from 41% (when applied as a stand-alone tool), up to 75% (when applied only to critical for failure resources), as compared to an existing state-of-the-art solution
  - without any accuracy degradation (average error about $3 \times 10^{-5} ^\circ C$).
Thank you