Reducing Power Density through Activity Migration

Seongmoo Heo, Kenneth Barr, and Krste Asanović

Computer Architecture Group, MIT CSAIL
Background

• Hot Spots
  – Rapid rise of processor power density
  – Uneven distribution of power dissipation
    • Blocks such as issue windows have more than 20x power density of less active block such as L2$
  – Reduced device reliability and speed, increased leakage current

• Existing Solutions
  – Packaging/cooling: high cost, not possible at laptop
  – Dynamic thermal management: performance loss
    • Total power dissipation must be reduced until all hot spots have acceptable junction temperature
Introduction

• **Activity Migration (AM)** to reduce power density
  – With AM, we spread heat by transporting computation to a different location on the die
  – If one unit heats past a temperature threshold, the computation is transferred to a second unit allowing the first to cool down
• AM for lowering temperature and power or for doubling maximum power dissipation at a given package
Die Thickness and Power Density

- **Two technology cases**
  - 180nm case: present, based on TSMC process
  - 70nm case: near future, based on BPTM process

- **Die thickness**
  - Most heat is removed through back of die
  - Thinning chips: 250um → 100um
  - Increasing lateral resistance

- **Power density**
  - Ideal scaling → constant power density
  - Vdd scale-down slowed, clock frequency increase accelerated due to deep pipelining → power density increase: 5W/mm² → 7.5W/mm²
Equivalent RC Thermal Model

- Equivalent RC Thermal Model:
  - temperature - voltage, power - current
- Thermal resistance: lateral resistance ignored
- Thermal capacitance: package capacitance modeled as a temperature source (isothermal point)
- Exponential dependence of leakage power on temperature modeled as voltage-dependent current source (P_leakage(Tj))
Benefits of Activity Migration

- **AM**: reduced temperature and power
- **AM + Perf-Pwr Tradeoff**: increased frequency and sustainable power

**Example**: laptop with limited heat removal
- Battery mode: AM Only: low temp, low leakage power → energy-efficient execution
- Plugged mode: AM+Perf-Pwr Tradeoff: more power, more performance → max. performance execution without raising die temperature
Activity Migration Model

- Activity Migration by turning on and off active power of hotspot and duplicated blocks (P_act1 and P_act2)
- Identical thermal resistance and capacitance
- Identical leakage power at same temperature
AM Only

Active Power

- $P_{base}$
- $0$
- $P_{act1}$
- $P_{act2}$

Temperature

- $T_{base}$
- Reduced Temperature
- $T_{iso}$
- $T_{j1}$
- $T_{j2}$

Migration Period

Time
AM + Perf-Pwr Tradeoff

Increased sustainable power by AM + Perf-Pwr Tradeoff

Migration Period
Migration Period: AM Only

Active Power
P_{base} 0
P_{act2} - short P_{act2} - long
Time

Temperature
T_{base} T_{iso}

Temp can be reduced till (T_{base}+T_{iso})/2

Migration Period

Time
Migration Period: AM + Perf-Pwr Tradeoff

Sustainable power can be increased till $2 \times P_{base}$
Effect of Migration Period

- Small migration period
  + More temperature drop (More power increase)
  - Greater CPI penalty
  - AM in hardware: Hardware overhead

- Large migration period
  + Smaller CPI penalty
  + AM in software: OS context swap
  - Less temperature drop (Less power increase)
Simulation Results: AM Only

- Reduced temperature $\rightarrow$ reduced leakage power
- Reduced latency due to increased drain current at low temperature is exploited by reducing $V_{dd}$ $\rightarrow$ reduced active power

<table>
<thead>
<tr>
<th>Migration period ($\mu$s)</th>
<th>180nm Case</th>
<th>70nm Case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1800</td>
<td>600</td>
</tr>
<tr>
<td>Temperature drop (K)</td>
<td>9.2</td>
<td>11.5</td>
</tr>
<tr>
<td>Leak power reduction (%)</td>
<td>29.6</td>
<td>35.3</td>
</tr>
<tr>
<td>Act power reduction (%)</td>
<td>3.7</td>
<td>7.6</td>
</tr>
</tbody>
</table>
Simulation Results: AM+Perf-Pwr Tradeoff

- Same temperature as baseline
- Perf-Pwr Tradeoffs: DVS, dynamic cache configuration modification, fetch/decode throttling, or speculation control
- DVS chosen for Perf-Pwr Tradeoff due to its simplicity

<table>
<thead>
<tr>
<th>Migration period (µs)</th>
<th>180nm Case</th>
<th>70nm Case</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1800</td>
<td>600</td>
</tr>
<tr>
<td>Freq increase (%)</td>
<td>10.5</td>
<td>14.1</td>
</tr>
<tr>
<td>Power increase (%)</td>
<td>56.8</td>
<td>79.5</td>
</tr>
</tbody>
</table>
**AM Architecture Configuration**

<table>
<thead>
<tr>
<th>I$ , ITLB, Branch Predictor</th>
<th>Issue Queue, Rename Table</th>
<th>Execution Units, Register File</th>
<th>D$ , DTLB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

- **Base:** block areas based on Alpha 21264 floorplan
- **Hotspot blocks:** execution units and register file
- **Pessimistic CPI penalties of AM**
  - Cycle penalty due to increased wire latency when sharing a block: e.g. Shared D$ $\rightarrow$ extra cycle to cache access time
  - Migration penalty: draining and copying
Performance Effects of AM

• Methodology
  • 4-wide 32-bit superscalar machine
  • SimpleScalar 3.0b
  • SPEC2000 benchmarks using SimPoints

• Migration Period
  • Short migration period chosen: 200K cycles
    (200µs for 180nm case and 60 µs for 70nm case)

Only 0~3% CPI penalty on average even at short migration period
## Effects of AM for Area and Net Perf

<table>
<thead>
<tr>
<th></th>
<th>180nm Case</th>
<th></th>
<th>70nm Case</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Conf</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
</tr>
<tr>
<td>Area</td>
<td>2.00</td>
<td>1.84</td>
<td>1.56</td>
<td>1.30</td>
</tr>
<tr>
<td>Speed</td>
<td>1.16</td>
<td>1.13</td>
<td>1.12</td>
<td>1.12</td>
</tr>
</tbody>
</table>

- Normalized to baseline, speed = clock freq / CPI

- **180nm Case**: conf. D achieves 12% performance gain with 30% area increase
- **70nm Case**: performance gain relatively small → AM only to cool down hot spots
- **Other issues**
  - Extra power for driving increased wire lengths
  - Migration triggering by thermal sensors rather than fixed migration periods
Conclusion

• Activity Migration (AM) was proposed to solve hotspot problem of modern microprocessors
• AM spreads heat by transporting computation to a duplicated block
• AM can be used in two ways
  1. AM only: low temperature, low leakage
  2. AM + Performance-Power Tradeoff: sustainable power and performance increase
• Dynamic fixed-period AM was evaluated on a superscalar machine
  – 12.7 degree temperature reduction
  – 12% clock frequency increase with 3% CPI penalty and 30% area increase
Acknowledgments

• Thanks to Christopher Batten, Ronny Krashinsky, Heidi Pan, and anonymous reviewers

• Funded by DARPA PAC/C award F30602-00-2-0562, NSF CAREER award CCR-0093354, and a donation from Intel Corporation.
BACKUP SLIDES
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Current Case</th>
<th>Future Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>250</td>
<td>100</td>
</tr>
<tr>
<td>K</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>1e6</td>
<td>1e6</td>
</tr>
<tr>
<td>A\textsubscript{die}</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>A\textsubscript{block}</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>PD\textsubscript{act}</td>
<td>5</td>
<td>7.5</td>
</tr>
<tr>
<td>PD\textsubscript{leak}</td>
<td>0.015</td>
<td>0.15</td>
</tr>
<tr>
<td>T\textsubscript{iso}</td>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>L</td>
<td>180</td>
<td>70</td>
</tr>
<tr>
<td>V\textsubscript{DD}</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td>NV\textsubscript{th0}</td>
<td>0.269</td>
<td>0.120</td>
</tr>
<tr>
<td>PV\textsubscript{th0}</td>
<td>-0.228</td>
<td>-0.153</td>
</tr>
</tbody>
</table>

* Transistor models: TSMC 180nm and BPTM 70nm processes
Equivalent RC Thermal Model

- $R_{\text{silicon, vertical}} = \frac{t}{k \times A_{\text{block}}}$
- $R_{\text{package, vertical}} = 120 \times \frac{t}{k} \times \frac{A_{\text{die}}}{A_{\text{block}}}$
- $R_{\text{total, vertical}} = (1 + 120 \times A_{\text{die}}) \times \frac{t}{k \times A_{\text{block}}}$
- $C_{\text{silicon}} = c \times t \times A_{\text{block}}$

*Empirical formula from 3D simulation results [Barcella02]

Exponential dependence of leakage power upon temperature modeled by voltage-dependent current source
Temperature Dependency of Leakage

- Leakage power
  - Significant part of total power
  - Exponential dependence upon temperature
  - Voltage-dependent current source

\[ P_{\text{leak}} = P_{\text{leak}110} \times e^{\beta(T_j-110)} \]
If period is small enough,
• **Halve** temp increase
• **Double** sustainable power
AM Simulation Results: AM + DVS

AM and DVS for various pingpong periods for the hot spot block (Current case)

DVS effects were modeled based on Hspice simulation of a 15-stage ring-oscillator
AM Simulation Results: AM + DVS

AM and DVS for various pingpong periods for the hot spot block (Future case)
Performance Effects of AM

- 4-wide 32-bit superscalar machine
- SimpleScalar 3.0b
- SPEC2000 benchmarks using SimPoints
- Short migration period chosen: 200K cycles (200µs for 180nm case and 60 µs for 70nm case)