Power and Temperature-Aware Clock Frequency and Thread Assignment in Multi-layer MPSoC

Kyungsu Kang*, Sungjoo Yoo** and Chong-Min Kyung*

* KAIST
** POSTECH
Contents

• Introduction
  – Challenges in multi-layer MPSoC
  – Problem definition and Preliminaries
  – Relate works
  – Motivational Example

• Proposed method
  – Temperature-slack based DVFS
  – Thread assignment exploiting memory-boundness

• Experimental result

• Conclusion
3D integration of MPSoC

Merits:
1. Small footprint
2. Short wire length
3. Heterogeneous integration
4. Wide bandwidth

Challenges:
1. Temperature
2. Yield
3. CAD support
Challenges of multi-layer MPSoC

Performance

Cooling cost

Leakage power

\[ P_l = A \cdot T^2 \cdot \exp\left( \frac{-B}{T} \right) \]

Reliability

\[ MTTF = A \cdot \exp\left( \frac{E_a}{k \cdot T} \right) \]
Objective of the research

- Developing temperature-aware power management methods (i.e., DVFS, thread assignment) to maximize instruction throughput in 3D multi-processor systems.
Challenges

• Different thermal characteristics compared with 2D systems

• Instantaneous (not steady-state) temperature analysis

• Consideration of workload characteristics (e.g., instructions per cycles, memory-boundness)

• Many systems with peak power constraint [ISCA05][Intel]


Problem definition

Given $N$ cores and $N$ threads to be assigned;

**Find** schedules of voltage/frequency and thread assignment on each core

**such that** total IPS ( = sum of IPS for all cores) is maximized

(IPS : instructions per second)

**subject to** satisfying both peak instantaneous power and peak instantaneous temperature constraints.
Thermal characteristics of 3D systems

- Heat flow
  - Heat is propagated vertically to the heat sink through other cores in between and dissipated at the heat sink.

Simplified thermal model [Zhu, TCAD08]

\[
R_{hs} = 1.22 \text{ K/W} \\
R_{inter} = 0.15 \text{ K/W} \\
R_{intra} = 2.44 \text{ K/W}
\]

\[
R_{intra} \approx 16 \cdot R_{inter}
\]

3D thermal characteristics

- Thermal coupling
  - Thinning of silicon layer makes strong mutual thermal coupling among vertical adjacent cores.

Thermal resistance: \[ R_{th} = \frac{H}{k \cdot A} \]

where
- \( H \): thickness
- \( k \): thermal conductivity
- \( A \): surface area

3D thermal characteristics

- Layer-dependent cooling efficiency
  - Cores near the heat sink have lower temperature than those far from the heat sink

\[
T_{2ss} = (P_2 + P_3) \cdot R_{hs} + T_{amb}
\]

Steady-state temperature:

\[
T_{3ss} = P_3 \cdot R_{inter} + T_{2ss}
\]

\[
T_{2ss} \leq T_{3ss} ; \text{ Equality holds if and only if } P_3 = 0.
\]

Cool job on Core 3
Hot job on Core 2

Source: Zhou, TPDS10

Instantaneous vs. steady-state temperature

- High-level temperature model [Liao, TCAD05]:

\[
T(t) = (T_{\text{init}} - (T_{\text{amb}} + R \cdot P)) \cdot e^{-t / R \cdot C} + R \cdot P + T_{\text{amb}} = (T_{\text{init}} - T_{\text{ss}}) \cdot e^{-t / R \cdot C} + T_{\text{ss}}
\]

where \( T_{\text{init}} (T_{\text{ss}}) \) is initial (steady-state) temperature of a core,
\( P \) is power consumption of a core,
\( R \) and \( C \) are thermal resistance and capacitance of a core, respectively.

**Driving temperature force**

Instantaneous temperature needs to be considered as thermal time constant of several hundreds of milliseconds is much longer than DVFS time step. [Skadron, TACO04]


Effect of memory-boundness in DVFS

- Execution time of an application:
  \[ t_{\text{ex}}(f_{\text{core}}) = \frac{w_{\text{comp}}}{f_{\text{core}}} + t_{\text{stall}} \]

  \( w_{\text{comp}} \): computation workload, \( f_{\text{core}} \): clock frequency of core
  \( t_{\text{stall}} \): stall time spent by core for external memory access

- Speedup (\( SU \)) for three programs in SPEC2000

\[
SU = \frac{t_{\text{ex}}(f_{\text{core}}^{\text{ref}} = 1.0\text{GHz})}{t_{\text{ex}}(f_{\text{core}}^{\text{new}})}
\]

Low \( SU \) is due to high memory-boundness
## Related works

<table>
<thead>
<tr>
<th>Previous works</th>
<th>Constraint</th>
<th>Temperature analysis</th>
<th>Memory-boundness</th>
<th>Solution type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Coskun, TVLSI08] [Zhang, ICCAD08]</td>
<td>Temperature</td>
<td>Instantaneous temperature</td>
<td>Not handled</td>
<td>Design-time</td>
</tr>
<tr>
<td>[Murali, CODES+ISSS07]</td>
<td>Temperature and power</td>
<td>Instantaneous temperature</td>
<td>Not handled</td>
<td>Design-time</td>
</tr>
<tr>
<td>[Annavaram, ISCA05] [Isci, Int. Symp. Microarchitecture06] [Bergamaschi, ASPDAC08]</td>
<td>Power</td>
<td>Steady-state temperature</td>
<td>Handled</td>
<td>Runtime</td>
</tr>
<tr>
<td>[Donald, ISCA06]</td>
<td>Temperature</td>
<td>Instantaneous temperature</td>
<td>Handled</td>
<td>Runtime</td>
</tr>
<tr>
<td>3D integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>[Zhao, APCCAS08]</td>
<td>Temperature and power</td>
<td>Instantaneous temperature</td>
<td>Not handled</td>
<td>Design-time</td>
</tr>
<tr>
<td>[Zhu, TCAD08]</td>
<td>Temperature</td>
<td>Steady-state temperature</td>
<td>Not handled</td>
<td>Runtime</td>
</tr>
<tr>
<td>Proposed</td>
<td>Temperature and power</td>
<td>Instantaneous temperature</td>
<td>Handled</td>
<td>Runtime</td>
</tr>
</tbody>
</table>
Motivational Example

• Preliminaries
  – Threads 1 and 2 are assigned to cores 1 and 2, respectively.
  – Each core has four frequency levels (i.e., 1.00, 1.33, 1.66, 2.00 GHz).
  – Constraints: $P_{\text{max}} = 52\text{W}$, $T_{\text{max}} = 63^\circ\text{C}$
Motivational Example
(Effect of considering instantaneous temperature)

Steady-state temperature-based DVFS
[Zhu, TCAD08]

Temperature slack-based
(Instantaneous temperature-based)
DVFS

8.8 % IPS improvement while keeping temperature constraint
Motivational Example

(Effect of task assignment)

Temperature slack-based DVFS

- **Time** (ms)
  - *T1 finish* @ 891.2 ms
  - *T2 finish* @ 647.6 ms

- **Frequency (GHz)**
  - Core 1
  - Core 2

- **Temperature (°C)**
  - *T_max = 63°C*
  - Core 1
  - Core 2

Temperature slack-based DVFS with different thread assignment

- **Time** (ms)
  - *T1 finish* @ 891.2 ms
  - *T2 finish* @ 647.6 ms

- **Frequency (GHz)**
  - Core 1
  - Core 2

- **Temperature (°C)**
  - *T_max = 63°C*
  - Core 1
  - Core 2

11.1 % IPS improvement while keeping temperature constraint

MPSoC Workshop, Gifu, 2010, Kyung
Temperature slack-based DVFS

- Two-step approach
  - Power budgeting based on per-core temperature slack
  - Optimal frequency assignment based on the assigned per-core power budget

Example of two-step approach:

- Power budget, $P_{max}$
  - 200 W
  - 22 W
  - 25 W
  - 30 W

Power budgeting among cores

Optimal frequency assignment for each core

- Core 1 @ 60°C → 1.6 GHz @ 21.6 W
- Core 2 @ 57°C → 1.6 GHz @ 21.6 W
- Core N @ 52°C → 2.0 GHz @ 27.7 W
Terminology

- Core-set: cores in the same horizontal position
- Top core: the core farthest from the heat sink within a core-set

\[
\text{Core-set } (i, j, k) = \text{core } (i, j, k) \times \text{Core layers} \\
\text{Top core } (4, 3) = \text{core } (4, 3, Z) \\
\text{Core-set } (4, 3) \\
N = X \cdot Y \cdot Z
\]
**Temperature slack-based DVFS**

- **Two-step approach**
  - Power budgeting among core-sets based on the temperature slack of top core in each core-set
  - Optimal frequency assignment of cores within a core-set based on the assigned power budget of core-set

Example of two-step approach:

- **Power budget, \( P_{max} \)**
  - 200 W
  - 60 W
  - 30 W

- **Power budgeting among core-sets**

- **Optimal frequency assignment within a core-set**
  - Core 1
    - 1.0 GHz @ 11.3 W
  - Core 2
    - 1.6 GHz @ 21.6 W
  - Core Z
    - 2.0 GHz @ 27.7 W
Reducing complexity in power budgeting

**Find** power budget of each core-set (\( P_{i,j}^{\text{core-set}} \))

**such that** total IPS (instructions per second) is maximized

**subject to** \( \forall i, \forall j; \; T_{i,j,z}(t) \leq T_{\text{max}} \) and \( \sum_{i=1}^{X} \sum_{j=1}^{Y} P_{i,j}^{\text{core-set}} \leq P_{\text{max}} \)

---

**Core-set**

Power budgeting among cores

Power budgeting among core-sets

MPSoc Workshop, Gifu, 2010, Kyung
Power budgeting among core-sets

- Power budgeting among core-sets in the form of assigning steady-state temperature

\[ \forall i, \forall j; \quad \frac{T_{\text{max}} - T_{i,j,Z}(t)}{T_{\text{ss}} - T_{i,j,Z}(t)} = C \quad \iff \quad \frac{\text{Temperature slack of core-set}}{\text{Temperature driving force of core-set}} = C \quad \text{... Eqn. (x)} \]

\[ T_{i,j,Z}^{\text{ss}} = R_{i,j}^{\text{core-set}} \cdot P_{i,j}^{\text{core-set}} + T_{\text{amb}} \]

where \( T_{i,j,Z}(t) \) is the temperature of top core \((i,j)\) at time \(t\)

\( T_{i,j,Z}^{\text{ss}} \) is the steady-state temperature of top core \((i,j)\)

\( R_{i,j}^{\text{core-set}} \) is the thermal resistance of core-set \((i,j)\)

\( C \) is a constant

Theorem: Performance is maximized when power is assigned to each core-set such that the equation (x) is satisfied.
Proof of power budgeting theorem (1)

\[
\frac{T_{\text{max}} - T_i(t)}{T_{i}^{ss} - T_i(t)} = C \iff \frac{T_{\text{max}} - T_i(t)}{dT_i(t)/dt} = C'
\]

\[
\therefore \frac{dT_i(t)}{dt} = (T_i^{ss} - T_i^{\text{init}}) \cdot e^{-t/RC} \cdot \frac{1}{RC} \propto T_i^{ss} - T_i^{\text{init}}
\]

\(C\) represents the time spent in each core to completely close the temperature slack remaining at time \(t\).

\[
\frac{T_{\text{max}} - T_i(t)}{dT_i(t)/dt} = C' \iff \frac{T_{\text{max}} - T_i(t)}{\eta \cdot (R_i \cdot P_i(t) + T_{\text{amb}} - T_i^{\text{init}})} = C'
\]

\[
\therefore \frac{T_{\text{max}} - T_i(t)}{K \cdot P_i(t) + L} = C' \quad \therefore T_i^{ss} = R_i \cdot P_i + T_{\text{amb}}
\]

\(dT_i(t)/dt\) linearly increases as power \((P_i)\) increases.
Proof of power budgeting theorem (2)

(a) Set power budget of each core such that both cores completely close the temperature slack at the same time

(b) Set power budget of each core such that Core 1 closes its temperature slack earlier than Core 2

Executed workload ($w$) of (b): (we assume that $P \propto f^\beta$)

\[
w = \sqrt[\beta]{P_1^{C^\prime} + \alpha \cdot (C^\prime - \Delta t)} + \sqrt[\beta]{P_2^{C^\prime} - \alpha \cdot (C^\prime - \Delta t)} + \sqrt[\beta]{P_{\text{max}}} \cdot \Delta t
\]

\[
= \sqrt[\beta]{P_1^{C^\prime} + \alpha \cdot (C^\prime - \Delta t)} + \sqrt[\beta]{P_{\text{max}} - P_1^{C^\prime} - \alpha \cdot (C^\prime - \Delta t)} + \sqrt[\beta]{P_{\text{max}}} \cdot \Delta t
\]
Proof of power budgeting theorem (3)

(a) Set power budget of each core such that both cores completely close the temperature slack at the same time

(b) Set power budget of each core such that Core 1 closes its temperature slack earlier than Core 2

Partial derivatives of executed workload ($w$) of (b):

$$\frac{dw}{d\Delta t} = -\beta \sqrt{P_1^{C'}} + \alpha - \beta \sqrt{P_{\text{max}} - (P_1^{C'} + \alpha)} + \beta \sqrt{P_{\text{max}}} < 0$$

$\Rightarrow \quad w$ is maximized when $\Delta t$ is zero. (QED) $\therefore \sqrt{x}$ is a concave function. ($2 < \beta < 3$)
Power budgeting among core-sets

• Constraints in power budgeting

\[ \forall i, \forall j; \quad T_{i,j,Z}(t + \Delta t) \leq T_{\text{max}} \quad (1) \]

\[ \sum_{i=1}^{X} \sum_{j=1}^{Y} P_{i,j}^{\text{core-set}} \leq P_{\text{max}} \quad (2) \]

where \( P_{i,j}^{\text{core-set}} \) is the power budget of core-set \((i, j)\)

\( \Delta t \) is the time interval for DVFS

\[ T_{i,j,Z}(t + \Delta t) = (T_{i,j,Z}(t) - T_{i,j,Z}^{ss}) \cdot e^{-\Delta t / RC} + T_{i,j,Z}^{ss} \]

Binary search is used to find the smallest \( C \) as defined below such that the two constraints (1) and (2) are both satisfied.

\[ \frac{T_{\text{max}} - T_i(t)}{T_i^{ss} - T_i(t)} = C \quad P_{i,j}^{\text{core-set}} \quad \text{and} \quad T_{i,j,Z}(t + \Delta t) \text{ are non-decreasing as } C \text{ decreases.} \]
Frequency assignment in a core-set

- Theorem: Following relations among frequencies within a core-set must hold to maximize the instruction throughput performance;

\[ \forall k; \quad R_k \cdot \frac{dP(f_k)}{df_k} = M \quad \text{where } M \text{ is a constant.} \]

Subject to \[ \sum_{k=1}^{Z} R_k \cdot P(f_k) + T_{amb} \leq T_{Z}^{ss} \]

\[ R_k = \sum_{l=1}^{k} r_l \]

\( P(f_k) \): power consumption of core running at \( f_k \)

Implication of the equation: With the same total amount of workload executed for all layers, the core on the layer farther from heat sink with larger \( R_k \) is assigned lower clock frequency (due to the upward concavity of \( dP(f_k)/df_k \)).
Proof of frequency assignment (1)

Find $f_k$ for all $k=1, 2, \ldots, Z$

where $f_k$ is clock frequency of the core located on layer $k$

such that $f_{total} = \sum_{k=1}^{Z} f_k$ is maximized (for maximum performance)

subject to $\sum_{k=1}^{Z} R_k \cdot P(f_k) + T_{amb} = T_Z^{ss}$ (for maximum frequency)
Proof of frequency assignment (2)

- Lagrange function: \( L(f_1, f_2, \ldots, f_Z, \lambda) = \sum_{k=1}^{Z} f_k + \lambda \left( \sum_{k=1}^{Z} R_k \cdot P(f_k) + T_{amb} - T_{ss} \right) \)

where \( \lambda \) is a Lagrange multiplier.

- Partial derivatives of the Lagrange function must be set to zero to maximize the objective;

\[
\frac{dL(f_1, f_2, \ldots, f_Z, \lambda)}{df_k} = 1 + \lambda \cdot R_k \cdot \frac{dP(f_k)}{df_k} = 0
\]

- Therefore, following relations among frequencies within a core-set must hold;

\( \forall k; \ R_k \cdot \frac{dP(f_k)}{df_k} = M \) where \( M \) is a constant.
Frequency assignment in a core-set

- Finding optimal discrete frequency levels

\[ f_1 = \text{the highest available clock frequency} \]

Determine the frequencies of remaining cores

\[ \forall k; \quad R_k \cdot \frac{dP(f_k)}{df_k} = M \]

Constraint check

\[ \sum_{k=1}^{Z} R_k \cdot P(f_k) + T_{amb} \leq T_{Z}^{ss} ? \]

Yes

No

MPSoC Workshop, Gifu, 2010, Kyung
Temperature-aware thread assignment

• Two objectives
  – Balancing temperatures among cores
  – Maximizing instruction throughput performance (i.e., total IPS)

Points to consider:
1) IPC (instructions per cycle)
2) Memory-boundness
Temperature-aware thread assignment

- Two-step approach
  - Thread-set assignment among core-sets to balance temperatures among core-sets
  - Thread assignment within a core-set to maximize IPS

Example of two-step approach (step 1):

Thread:  1  2  3  ...  N

Thread-set:  1  2  ...  Z
  1  2  ...  Z
  ...  ...  ...
  1  2  ...  Z

Core-set

Heat sink

MPSoC Workshop, Gifu, 2010, Kyung
Thread-set assignment among core-sets

• Objective of thread-set formation
  – Balancing IPC sums among thread-sets

• Procedure of thread-set formation
  – 1) Assume $X \cdot Y$ empty sets which can store maximally $Z$ threads.
  – 2) Sort all threads according to the descending order of IPC.
  – 3) Put the thread with the highest IPC into a set with the lowest sum of IPCs.
  – 4) Repeat 3) until all threads are assigned to one of $X \cdot Y$ sets.
Thread-set assignment among core-sets

Example of thread-set formation: forming three thread-sets when $Z = 2$

- IPC sum: 1.2
- IPC sum: 1.1
- IPC sum: 1.0
Thread-set assignment among core-sets

• Procedure
  1. Sort all thread-sets according to the ascending order of IPC sum.
  2. Assign the thread-set with the lowest IPC sum to the core-set with the highest temperature of top core.
  3. Repeat 2 until all the thread-sets are assigned.

Relation between IPC and switching power: \( P_s (V_{dd}, f) = C_s \cdot V_{dd}^2 \cdot f \propto IPC \)

Example of IPC sum:

IPC sum: 0.9 + 0.3 = 1.2
Temperature-aware thread assignment

• Two-step approach
  – Thread-set assignment among core-sets to balance temperatures among core-sets
  – Thread assignment within a core-set to maximize IPS

Example of two-step approach (step 2):
Thread assignment within a core-set

**Procedure**

1. Sort threads in a core-set according to the ascending order of SU.
2. Assign the thread with the lowest SU to the core farthest from the heat sink.
3. Repeat 2 until all threads in a core-set are assigned.

\[
\frac{T_{\max} - T_i(t)}{T_{i}^{ss} - T_i(t)} = C \quad \iff \quad \text{Higher temperature slack (i.e., lower core temperature) allows the assignment of larger power budget (i.e., higher voltage/frequency), and, therefore, task with larger SU.}
\]

Higher temperature slack (i.e., lower core temperature) allows the assignment of larger power budget (i.e., higher voltage/frequency), and, therefore, task with larger SU.
Experimental setup

Simulation environment

<table>
<thead>
<tr>
<th>Target processor</th>
<th>Intel 65-nm Merom processor</th>
<th>[Sakran, ISSCC07]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature estimation</td>
<td>3D grid-based temperature model</td>
<td>[Huang, TVLSI06]</td>
</tr>
<tr>
<td>Power estimation</td>
<td>Switching power + temperature-aware leakage power</td>
<td>[W. Liao, TCAD05]</td>
</tr>
<tr>
<td>Performance profiling</td>
<td>Performance profiling API on Intel Core2 processor in LW25 laptop</td>
<td>PAPI</td>
</tr>
</tbody>
</table>

Thermal characteristics [Coskun, DATE09]

<table>
<thead>
<tr>
<th>Layer</th>
<th>Thermal conductance (W/mK)</th>
<th>Heat capacitance (J/m³K)</th>
<th>Thickness (µm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat sink</td>
<td>400.0</td>
<td>3.55E+6</td>
<td>6,900</td>
</tr>
<tr>
<td>Heat spreader</td>
<td>400.0</td>
<td>3.55E+6</td>
<td>1,000</td>
</tr>
<tr>
<td>TIM</td>
<td>4.0</td>
<td>4.00E+6</td>
<td>20</td>
</tr>
<tr>
<td>Core / L2 cache</td>
<td>100.0</td>
<td>1.75E+6</td>
<td>150</td>
</tr>
<tr>
<td>Interlayer</td>
<td>4.0</td>
<td>4.00E+6</td>
<td>20</td>
</tr>
</tbody>
</table>

ETC.

<table>
<thead>
<tr>
<th>Frequency range (4 steps)</th>
<th>1 ~ 2 GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>∆t (DVFS time interval)</td>
<td>5 ms</td>
</tr>
<tr>
<td>DVFS overhead</td>
<td>10 µs [Lee, Tcomput.10]</td>
</tr>
<tr>
<td>Thread migration overhead</td>
<td>1 ms [Coskun, DATE09]</td>
</tr>
<tr>
<td>∆l (thread assignment interval)</td>
<td>100 ms</td>
</tr>
<tr>
<td>$P_{sleep}$</td>
<td>2 W [datasheet]</td>
</tr>
<tr>
<td>Maximum temperature</td>
<td>70 °C</td>
</tr>
<tr>
<td>Maximum power</td>
<td>200 W</td>
</tr>
</tbody>
</table>
Experimental setup

Comparison with existing solutions

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Power budgeting</th>
<th>Thread assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DI_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC and 2D floorplan-awareness</td>
</tr>
<tr>
<td>3DI_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC and 3D floorplan-awareness</td>
</tr>
<tr>
<td>3DIS_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC, SU and 3D floorplan-awareness</td>
</tr>
<tr>
<td>3DIS_IT (proposed)</td>
<td>Instantaneous temperature analysis</td>
<td>IPC, SU and 3D floorplan-awareness</td>
</tr>
</tbody>
</table>

Benchmark combinations (SPEC2000)

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPC</td>
<td>hipc, lipc, mipc</td>
</tr>
<tr>
<td>Speed up (SU)</td>
<td>hm, lm, mm</td>
</tr>
</tbody>
</table>

Ex.) hipc-hm: combination of applications with high IPC and high memory-boundness
Experimental results

IPS result of each combination of applications:

1) 2DI_SST → 3DI_SST; 8.0 % IPS improvement
2) 3DI_SST → 3DIS_SST; 8.5 % IPS improvement
3) 3DIS_SST → 3DIS_IT; 14.5 % IPS improvement

<table>
<thead>
<tr>
<th>Application</th>
<th>Analysis Type</th>
<th>Awareness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2DI_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC and 2D</td>
</tr>
<tr>
<td>3DI_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC and 3D</td>
</tr>
<tr>
<td>3DIS_SST</td>
<td>Steady state temperature analysis</td>
<td>IPC, SU and 3D</td>
</tr>
<tr>
<td>3DIS_IT (proposed)</td>
<td>Instantaneous temperature analysis</td>
<td>IPC, SU and 3D</td>
</tr>
</tbody>
</table>
Experimental results

 IPS result of each combination of applications:

1) Utilizing thermal characteristics of 3D floorplan and 2) memory-boundness of each applications

3) Aggressive power budgeting by exploiting the instantaneous temperature analysis

Reasons for instruction throughput improvement:

1) Utilizing thermal characteristics of 3D floorplan and 2) memory-boundness of each applications

3) Aggressive power budgeting by exploiting the instantaneous temperature analysis
Experimental results

EPI (energy per instruction) result:

Reasons for EPI increase:
Proposed method assigns higher frequencies to cores through aggressive power budgeting (gives average 24% instruction throughput improvement)

Reasonable overhead in energy efficiency

2DI_SST $\rightarrow$ 3DIS_IT; 3.4 % EPI increase
3DI_SST $\rightarrow$ 3DIS_IT; 1.3 % EPI increase
Experimental results

Computational time measured from LG xnote LW25 laptop running 2GHz:

<table>
<thead>
<tr>
<th>Method</th>
<th>Measured time</th>
<th>Time interval to invoke</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVFS</td>
<td>110 µs</td>
<td>5 ms</td>
</tr>
<tr>
<td>Thread assignment</td>
<td>3 µs</td>
<td>100 ms</td>
</tr>
</tbody>
</table>

# of clock frequency changes per second:

Average 61 clock frequency changes during 1s in 3DIS_IT.

The overhead of DVFS is negligible.
Summary

• Temperature-aware power management in multi-layer MPSoC
  – Dynamic voltage frequency scaling based on temperature slack
    • Power budgeting among core-sets based on the temperature slack of top core in each core-set
    • Optimal frequency assignment of cores within a core-set based on the assigned power budget of core-set
  – Temperature-aware thread assignment
    • Thread-set assignment among core-sets to balance temperatures among core-sets
    • Thread assignment within a core-set to maximize IPS
  – Experimental result shows 41% (24% on average) IPS improvement compared with existing methods