SysteMorph: AnSoC Framework for Adaptive Dynamic Optimization Systems

Noori, Hamid
Department of Informatics, Kyushu University

Eshima, Kazuhito
Department of Informatics, Kyushu University

Fujii, Yousuke
Department of Informatics, Kyushu University

Yoshida, Makoto
Fukuoka Industry, Science & Technology Foundation

http://hdl.handle.net/2324/6231
SysteMorph: An SoC Framework for Adaptive Dynamic Optimization Systems

Hamid Noori †, Kazuhito Eshima†, Yousuke Fujii†, Makoto Yoshida††, Takeshi Soga††, Norifumi Yoshimatsu††, Takanori Hayashida† & Kazuaki Murakami†

†Department of Informatics, Kyushu University
††Fukuoka Industry, Science & Technology Foundation
E-mail: †arch-systemmorph@c.csce.kyushu-u.ac.jp, ††ngarch@fleets.jp
Abstract

This research investigates a possible architecture to adaptive dynamic optimization systems. In this architecture, the running application is monitored and high frequently executed parts of the code are detected. Then, these parts of code are optimized, according to the architecture of embedded hardware accelerator. To be able to use the hardware accelerator, the new binary code is rewritten. IPFlexDAP/DNA-HP was the target of our prototyping to validate the concept. The proposed architecture for SoC implementation utilizes dynamic software pipelining technique for optimization and a simplified 8-way VLIW as the accelerator. Some preliminary performance evaluations show speedup.
Concept of SysteMorph

Before Optimization

Online Profiling
Target Programs
SysteMorph Software
Instruction Execution
Processor
Profiler
Programmable Engine
ISA

After Optimization

Hints for Optimization
Binary Rewriting
Target Programs
SysteMorph Software
Instruction Execution
Processor
Profiler
Programmable Engine
ISA

Hardware Reconfiguration
Mode Control
Adaptive HW/ISA/SW Co-optimization

Before Optimization

Online Profiling
Target Programs
SysteMorph Software
Instruction Execution
Processor
Profiler
Programmable Engine
ISA

After Optimization

Hints for Optimization
Binary Rewriting
Target Programs
SysteMorph Software
Instruction Execution
Processor
Profiler
Programmable Engine
ISA

Hardware Reconfiguration
Mode Control
Adaptive HW/ISA/SW Co-optimization

ERSA'05 3
SysteMorph Synthesis Flow on DAP/DNA-HP

Flow for synthesis

- Hot path information
- Renaming
- Optimization
- Generate Control/data flow graph
- Timing matching
- Mapping
- Place and Route

Flow for instruction generation

- Generate instructions to configure DNA
- Generate instructions to control DNA

DNA (Dynamic reconfigurable Logic)

External Memory

DAP (32bit RISC Processor)

Code for DNA configuration
Execution Flow on DAP/DNA-HP

Application Program (before optimization)
- Ld, Add, Cmp, Br, Add
- Branch taken
- Branch not taken
- Branch Exception
- SysteMorph Program
  - Online Profile

Application Program (After Optimization)
- Br NOP NOP Br Add
- Instructions to control and configure DNA
- Activate DNA then waiting for completion
- Generate DNA instruction to control and write DNA configuration instruction
- Online Synthesis
- Binary Rewriting

Online Profile detected
- Online Profile
- Binary Rewriting
Branch History Method

- **Branch History Table**

<table>
<thead>
<tr>
<th>Branch Address</th>
<th>Branch Target Address</th>
<th>Number of Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abottom</td>
<td>Btop</td>
<td>300</td>
</tr>
<tr>
<td>Abottom</td>
<td>Ctop</td>
<td>200</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Number of Backward Branch > Threshold**

- **Find Hot Path**
  A path which has more frequently taken is chosen as a Hot Path.

\[
\text{Atop} : \text{Head Address of Basic Block A} \\
\text{Abottom} : \text{End Address of Basic Block A}
\]
SysteMorph with VLIW accelerator

- **CPU**
  - Simple RISC processor

- **VLIW based accelerator**
  - Instruction RAM
  - Compute intensive execution unit

- **H/W profiler assist**
  Store Branch history information
  - Branch Target Address
  - Branch Instruction Address
  - Number of Taken/Not Taken
Dynamic Trace Based Software Pipelining

Applying Software Pipelining

Compensation code

Hot Path

Original loop

After applying dynamic software pipelining

ERSA'05
Hot Paths
Preliminary Performance Evaluation

<table>
<thead>
<tr>
<th>N (way)</th>
<th>Int</th>
<th>FP</th>
<th>Ld/St</th>
<th>Br</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exp. 1</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Exp. 2</td>
<td>8</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Exp. 2</td>
<td>12</td>
<td>5</td>
<td>1</td>
<td>5</td>
</tr>
</tbody>
</table>

For dependency consideration, the speedup is calculated as follows:

- No Dependency
- Considering Dependency