Design Space Exploration for Hardware/Software Codesign of Multiprocessor Systems

A. Baghdadi, N-E. Zergainoh, W. Cesario, T. Roudier†, A.A. Jerraya

TIMA Laboratory - Grenoble France
† Arexsys, R&D - Meylan France
Outline

- Introduction
- Estimation/Exploration Methodology
- Application Example
- Conclusion
Motivations (1)

- Application Performance vs Multiprocessor Architectures
- New System Design Methods: Hw/ Sw Codesign
- Design Space Exploration
- Huge Number of Architectural Solutions
  - Example: $n=4$ tasks, $p=3$ kinds of technologies

$$309 \text{ Solutions} \quad \text{(not considering communication !)}$$

This number increases exponentially with $n$ and $p$
Motivations (2)

For each architecture, synthesis and low-level cosimulation takes days.

System-level simulation is $10^4$ times faster than at RT level.

Find an efficient estimation/exploration methodology.

System-level simulation is $10^4$ times faster than at RT level.

Find an efficient estimation/exploration methodology.
Objective

- To find at the “system level” an efficient estimation methodology

SDL Specification → Performance Estimation → Architectural Choices → Architecture exploration loop → Results: architectures performances
State of the Art

Existing works in codesign

- Mono-processor target architecture
  - PMOSS
  - COSYMA
  - LYCOS

- Multiprocessor target architecture
  - SpecSyn
  - POLIS
  - T-Y. Yen and W. Wolf
Contribution

- Provide an accurate performance estimation method enabling design space exploration in the case of Hw/Sw codesign

- Handles complex multiprocessor architectures

- Validated by a significant application example: a RAC system
Outline

- Introduction
- Estimation/ Exploration Methodology
- Application Example
- Conclusion
Methodology Overview

- Optimal trade-off between speed and accuracy

- Speed: system-level simulation
  - Lacks accurate timing information

- Solution: use a back-annotation approach
  The analysis of some implementations at RTL allows us to extract all timing elements needed for performance estimation of all feasible implementations
Global Flow

1. **System Level**
   - **SDL specification**
   - **Step 2**
     - **Back-annotation**
     - **Time-annotated SDL specification**

2. **RT Level**
   - **Time computation of basic elements**
   - **Step 1**
     - **Timing information**

3. **Step 3**
   - **Architecture modeling**

4. **Step 4**
   - **System level simulation**
     - **Results: architectures performances**
   - **Architecture exploration loop**

The diagram illustrates the flow of information and processes from SDL specification to system level simulation, including back-annotation and time computation.
Basic Elements

- Computation
- Communication

A **Basic Block** (BB) is a sequence of operations without control instructions nor communication.
Time Computation of Basic Blocks

SDL specification

Ident. & labeling of all BB

MUSIC: Hw no comm.

MUSIC: Sw no comm.

C Code

VHDL RTL

BB ident. & time compute

BB delays for Hw

Techno. 1

BB delays on µP1

µP1 asm

BB ident. & time compute

Techno. n

BB delays on µPn

µPn asm

BB ident. & time compute

Delays of all BBs for all technologies

Techno. n
Communication

- Communication may be internal or external
- Modeling the communication time and synchronization into three kinds of delays:
  - Interface initialization time (external comm.) $T_{\text{Startup}}$
  - Data transmission time $T_{\text{Trans}}$
  - Synchronization time $T_{\text{Synchro}}$

\[ T_{\text{Comm}}(n) = \lambda T_{\text{Startup}} + T_{\text{Trans}}(n) + T_{\text{Synchro}} \]

$\lambda$ stands for 0/1 according to internal/external comm. resp.

$n$ is the amount of data

- Same flow as BB for calculation of $T_{\text{Startup}}$ and $T_{\text{Trans}}$
- Build a library of communication delays for reuse
Back-Annotation & Arch. Modeling

- Geodesim: extra annotations for the assessment of performance and architecture modeling (DELAY, NODE, PRIORITY)

```plaintext
SYSTEM sys;

BLOCK b1 COMMENT '#NODE';
SIGNALROUTE b1_b2.1 rendez-vous
FROM p1 TO ENV WITH value1;
...
PROCESS p1 COMMENT '#PRIORITY (1)';
...
TASK a := b*c COMMENT '#DELAY ((5+mult1+3)*p_8051)';
...
ENDPROCESS;
PROCESS p2;
...
ENDPROCESS;
ENDBLOCK;

BLOCK b2;
...
PROCESS p3;
...
PROCESS p4;
...
ENDBLOCK;
ENDSYSTEM;
```
System-level simulation

- Executing the Annotated SDL specification using GEODESIM simulator

- Defining the same test-bench for all architectures that we have to compare
Outline

- Introduction
- Estimation/ Exploration Methodology
- Application Example
- Conclusion
Results

<table>
<thead>
<tr>
<th>Archi_1</th>
<th>Host</th>
<th>PID</th>
<th>M1</th>
<th>M2</th>
<th>Performance (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80C51</td>
<td>VHDL</td>
<td>ST10</td>
<td>1750</td>
<td>1947</td>
<td></td>
</tr>
<tr>
<td>Archi_2</td>
<td>ST10</td>
<td>VHDL</td>
<td>ST10</td>
<td>1735</td>
<td>1936</td>
</tr>
<tr>
<td>Archi_3</td>
<td>ST10</td>
<td>VHDL</td>
<td>ST10</td>
<td>ST10</td>
<td>1550</td>
</tr>
<tr>
<td>Archi_4</td>
<td>VHDL</td>
<td>ST10</td>
<td>1350</td>
<td>1375</td>
<td></td>
</tr>
<tr>
<td>Archi_5</td>
<td>VHDL</td>
<td></td>
<td>250</td>
<td>256</td>
<td></td>
</tr>
</tbody>
</table>

Legend:
- Our estimation approach
- RT-level cosimulation
Results Analysis

Error rate of the estimation approach

<table>
<thead>
<tr>
<th>Architectures</th>
<th>Archi_1</th>
<th>Archi_2</th>
<th>Archi_3</th>
<th>Archi_4</th>
<th>Archi_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Rate</td>
<td>10.1%</td>
<td>10.4%</td>
<td>1.7%</td>
<td>1.8%</td>
<td>2.3%</td>
</tr>
</tbody>
</table>

- Precision lack when the processes of the two motors share the same processor
- With a predefined scenario, error rate drops from 10% to about 3%
- Simulation speed: at system level, sim. is $10^4$ times faster than at RT level
- Limitations:
  - Advanced microprocessors (Pipeline, internal parallelism, … )
  - Compilers optimizations
  - Only one performance factor is considered
Outline

- Introduction
- Estimation/ Exploration Methodology
- Application Example
- Conclusion
Conclusion

- Development and validation of a new estimation/exploration methodology for hardware/software codesign
- Deal with complex multiprocessor architectures
- Excellent trade-off between speed and accuracy
- Validated on MUSIC codesign flow
- Easy to automate and extendable to cover other performance factors