METAMOC: Modular Execution Time Analysis Using Model Checking

Mads Chr. Olesen <mchro@cs.aau.dk>
joint work with
Andreas Engelbrecht Dalsgaard, Martin Toft,
René Rydhof Hansen, Kim Guldstrand Larsen

Aalborg University

July 6th 2010
Overview of METAMOC

Timed automata models for hardware components and process functions:

Abstract process model and value analysis

Abstract hardware model with caching and pipelining

WCET 42 cycles
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

disassemble (objdump, Dissy)

generate (Assembly-to-UPPAAL)

Control Flow Graph (UPPAAL model)

combine

Caches (UPPAAL models)

value analysis (WALi)

Complete model (UPPAAL model)

model check (UPPAAL)

WCET
Overview of METAMOC

---

**Annotated executable**

- **Disassemble** (objdump, Dissy)
- **Assembly**
  - **Value analysis** (WALi)

**Control Flow Graph (UPPAAL model)**

- **Generate** (Assembly-to-UPPAAL)

**Pipeline (UPPAAL model)**

- **Combine**

**Main memory (UPPAAL model)**

- **Complete model (UPPAAL model)**

**Cache specifications**

- **Generate** (cache-gen)

**Caches (UPPAAL models)**

- **Model check (UPPAAL)**

**WCET**
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

disassemble (objdump, Dissy)

Assembly

value analysis (WALi)

Control Flow Graph (UPPAAL model)

combine

Complete model (UPPAAL model)

model check (UPPAAL)

WCET
Overview of METAMOC

- Annotated executable (UPPAAL model)
- Disassemble (objdump, Dissy)
- Assembly
- Value analysis (WALi)
- Control Flow Graph (UPPAAL model)
- Generate (Assembly-to-UPPAAL)
- Pipeline (UPPAAL model)
- Main memory (UPPAAL model)
- Caches (UPPAAL models)
- Generate (cache-gen)
- Cache specifications
- Combine
- Complete model (UPPAAL model)
- Model check (UPPAAL)
- WCET
Overview of METAMOC

1. Annotated executable
2. Pipeline (UPPAAL model)
3. Main memory (UPPAAL model)
4. Cache specifications

- disassemble (objdump, Dissy)
- generate (Assembly-to-UPPAAL)
- control flow graph (UPPAAL model)
- combine
- complete model (UPPAAL model)
- model check (UPPAAL)

- Assembly
- value analysis (WALi)
- Caches (UPPAAL models)
- WCET
Overview of METAMOC

Annotated executable  
\[\xrightarrow{\text{disassemble (objdump, Dissy)}}\]  
Assembly  
\[\xrightarrow{\text{value analysis (WALi)}}\]  

Pipeline (UPPAAL model)  
\[\xrightarrow{\text{generate (Assembly-to-UPPAAL)}}\]  
Control Flow Graph (UPPAAL model)  
\[\xleftarrow{\text{combine}}\]  
Complete model (UPPAAL model)  
\[\xrightarrow{\text{model check (UPPAAL)}}\]  
WCET  

Main memory (UPPAAL model)  
\[\xrightarrow{\text{generate (cache-gen)}}\]  
Cache specifications  

Complete model (UPPAAL model)  
Caches (UPPAAL models)
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

Disassemble (objdump, Dissy)

Assembly

Control Flow Graph (UPPAAL model)

Combine

Caches (UPPAAL models)

Model check (UPPAAL)

WCET

Value analysis (WALi)
Overview of METAMOC

- Annotated executable
- Pipeline (UPPAAL model)
- Main memory (UPPAAL model)
- Cache specifications
- disassemble (objdump, Dissy)
- generate (Assembly→UPPAAL)
- generate (cache→gen)
- Assembly
- Control Flow Graph (UPPAAL model)
- Caches (UPPAAL models)
- value analysis (WALi)
- combine
- Complete model (UPPAAL model)
- model check (UPPAAL)
- WCET
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

disassemble (objdump, Dissy)

generate (Assembly-to-UPPAAL)

generate (cache-gen)

Assembly

Control Flow Graph (UPPAAL model)

Value analysis (WALi)

combine

Caches (UPPAAL models)

Complete model (UPPAAL model)

model check (UPPAAL)

WCET
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

disassemble (objdump, Dissy)

generate (Assembly-to-UPPAAL)

generate (cache-gen)

Assembly

Control Flow Graph (UPPAAL model)

Caches (UPPAAL models)

value analysis (WALi)

combine

Complete model (UPPAAL model)

model check (UPPAAL)

WCET
Overview of METAMOC

Annotated executable

Pipeline (UPPAAL model)

Main memory (UPPAAL model)

Cache specifications

disassemble (objdump, Dissy)

generate (Assembly-to-UPPAAL)

generate (cache-gen)

Assembly

Control Flow Graph (UPPAAL model)

combine

Complete model (UPPAAL model)

model check (UPPAAL)

WCET
Current Work

- Support for pipelines
  - ARM9TDMI
  - ARM7TDMI
  - ATMEL AVR 8-BIT
- Support for instruction/data caches
  - Automatically generated
  - LRU/FIFO replacement policy
- Value analysis for predicting memory accesses
  - Implemented using Weighted Push-Down Systems
  - Inter-procedural
  - Currently syntactic constant-propagation
- Timing anomalies cannot be (consistently) handled
  - Experiments with caches are with LRU caches, not FIFO as on the real ARM9
Modelling in METAMOC

Overview of the ARM9 automata

- Main memory TA
- Instruction cache TA
- Data cache TA
- Decode stage TA
- Execute stage TA
- Memory stage TA
- Writeback stage TA
- Fetch stage TA
- Process function TAs

Synchronisation
Dependency through code
Modelling in METAMOC

Sketch of the function automata for a process
Modelling in METAMOC

ARM9TDMI pipeline
- Fetch stage
- Decode stage
- Execute stage
- Writeback stage

ARM920T Caches
- Instruction cache
- Data cache
- Main memory

ARM9 overview and sketch of pipeline automata
UPPAAL

Introduction
Modelling Approach
UPPAAL, explained
Experiments
Future Work
UPPAAL
Overview

sup: cyclecounter

Query

sup: cyclecounter

i

sup: cyclecounter \leq 13729

Comment

Status

Established direct connection to local server.

(Academic) UPPAAL version 4.1.3 (rev. 4410), September 2009 -- server.

Disconnected.

Established direct connection to local server.

(Academic) UPPAAL version 4.1.3 (rev. 4410), September 2009 -- server.
UPPAAL Zones

1 ≤ x ≤ 4
1 ≤ y ≤ 3

delays to

1 ≤ x, 1 ≤ y
-2 ≤ x - y ≤ 3

conjuncts to

3 ≤ x, 1 ≤ y
-2 ≤ x - y ≤ 3

projects to

3 ≤ x, y = 0
UPPAAL Zones

- Delay is cheap - large zones
  - Resilient to different memory wait delays
- Many small steps expensive - smaller zones
- Zones can be collapsed, overapproximation
Eliminating non-determinism

- Since no timing anomalies, cut down on the number of distinct paths as much as possible
- Pigeonhole optimisations
  - Iterate loops the maximum number of times
  - Don’t forward jump if path is subset of not jumping
- “Executing more code increases the execution time”
- Can be disabled if timing anomalies present
## Experiments

- Evaluation using WCET benchmark programs from Mälardalen Real-Time Research Centre
  - Applicability
  - Performance
- Discarded a number of programs
  - Floating point operations handled by software routines
  - Dynamic jumps
  - Some programs do not compile for our architectures
- 21 programs for ARM and 19 programs for AVR
- Manually annotated loop bounds
Experiments

<table>
<thead>
<tr>
<th>ARM9 w. LRU caches, 21 benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysable without caches</td>
</tr>
<tr>
<td>Analysable with instruction cache</td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
</tr>
<tr>
<td>Manual modification of instruction cache size</td>
</tr>
<tr>
<td>Analysable with data and instruction cache</td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
</tr>
<tr>
<td>Manual modification of data cache size</td>
</tr>
<tr>
<td>Manual syntax fix of model</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ATMEL AVR 8-bit, 19 benchmarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analysable</td>
</tr>
<tr>
<td>Unanalysable, state space explosion</td>
</tr>
</tbody>
</table>

Relative improvement in WCET for ARM9.

Analysis times in minutes for AVR and ARM9.
Future Work

- Improvements in model checker technology
  - Our models atypical: more deterministic, longer paths, larger
  - Summarizing long deterministic paths - “short-cuts”
  - Parallel/Distributed model checking
  - Guiding the search - A*

- Data sensitivity/flow facts
  - Track values of registers in model

- Timing anomalies
  - Introduces more non-determinism
  - Improving model checker technology

- Schedulability instead of WCET analysis
  - SARTS project has done this for Java bytecode on the JOP processor
Thank you for your attention!

Questions?

http://metamoc.dk
Introduction
- Overview of METAMOC
- Current Work

Modelling Approach
- Modelling in METAMOC
- Model Checking using UPPAAL

UPPAAL, explained
- UPPAAL Zones
- Eliminating non-determinism

Experiments
- Experiments

Future Work
- Future Work