

WCET Analysis in Shared Resources Real-Time Systems with TDMA Buses

Claire Maiza

in collaboration with:

Hamza Rihani, Matthieu Moy and Sebastian Altmeyer
published in RTNS 2015

OVSTR, 11/06/2018



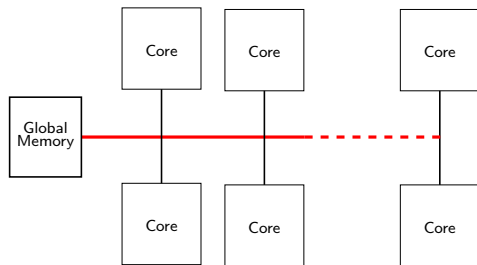
Precise static WCET analysis: combining path and TDMA analyses



Main idea

- TDMA causes no direct interference
- Task execution time does not depend on what is executed on other cores
- New analysis that combines:
 - ▶ a precise analysis on when memory accesses happen (path analysis)
 - ▶ a precise analysis on tdma slot (bus analysis)
- Use of SMT encoding

- 1 Context: TDMA bus
- 2 Motivation on an example
- 3 Our Contribution
- 4 Performance Evaluation
- 5 Conclusion and Discussion

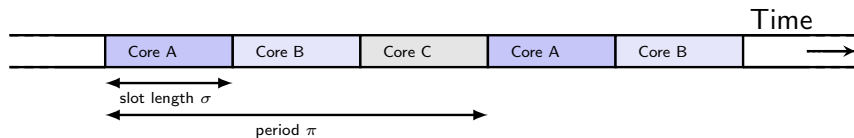


- Shared resources: contention, access delays
- Time Division Multiple Access (TDMA) arbitration policy
 - ▶ Time triggered policy
 - ▶ Assign fixed time slots to each core

Time Division Multiple Access



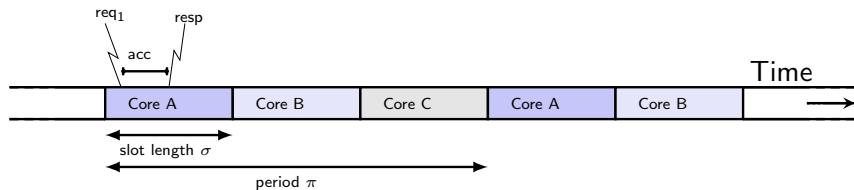
Time Division Multiple Access



Time Division Multiple Access



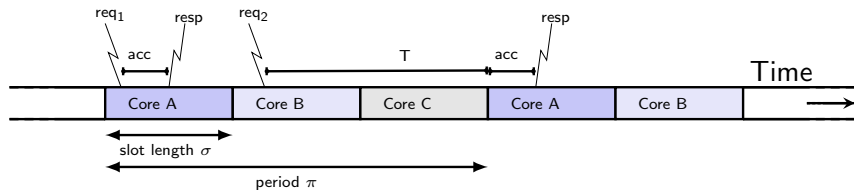
Core A viewpoint:



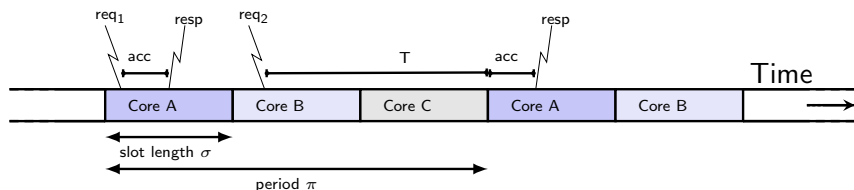
Time Division Multiple Access



Core A viewpoint:

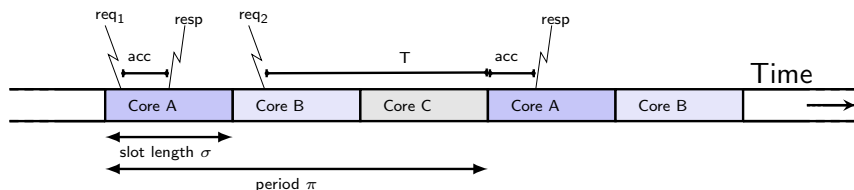


Core A viewpoint:



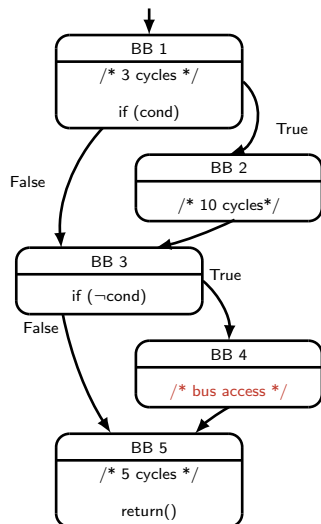
- acc execution time of a granted request
 - T waiting time to grant requests $\in [0, \pi - (\sigma - acc)]$
- Pessimistic approach:** $Worst-Case(T) = \pi - (\sigma - acc)$

Core A viewpoint:

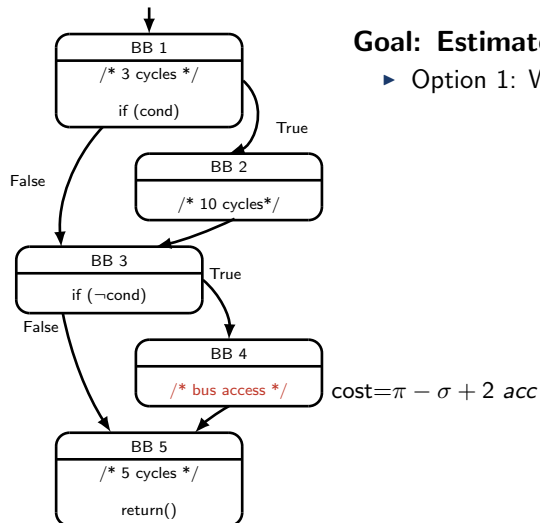


- *acc* execution time of a granted request
- *T* waiting time to grant requests $\in [0, \pi - (\sigma - acc)]$
- **Pessimistic approach:** $Worst-Case(T) = \pi - (\sigma - acc)$
- Offset relative to the TDMA period:

$$Offset(req) = time_instant(req) \bmod \pi$$

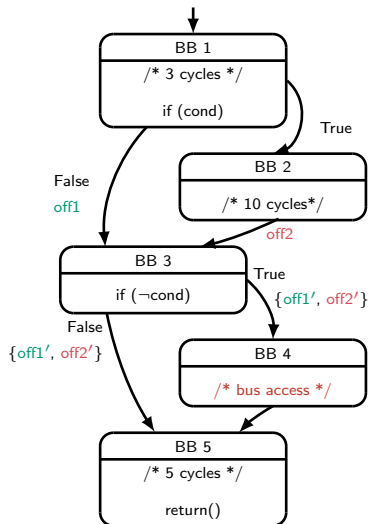


Goal: Estimate the WCET



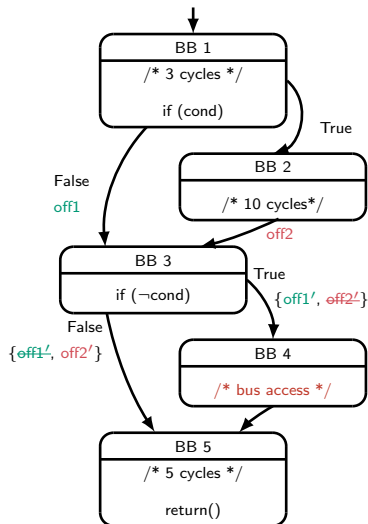
Goal: Estimate the WCET

- ▶ Option 1: Worst-case everywhere.



Goal: Estimate the WCET

- ▶ Option 1: Worst-case everywhere.
- ▶ Option 2: Capture all possible offsets



Goal: Estimate the WCET

- ▶ Option 1: Worst-case everywhere.
- ▶ Option 2: Capture all possible offsets
- ▶ Option 3: Feasible Path Analysis

TDMA Bus Analysis

Kelter et al., RTS'14

Chattopadhyay et al., SCOPES'10

Feasible Path Analysis with SMT

Henry et al. LCTES'14

TDMA Bus Analysis

Kelter et al., RTS'14

Chattopadhyay et al., SCOPES'10

Feasible Path Analysis with SMT

Henry et al. LCTES'14

Contribution:

Compute WCET by encoding the semantics and shared resource accesses into an optimization problem (SMT)

- Bounded Model Checking
 - ▶ Encode the semantics into a Satisfiability Modulo Theory problem

- Bounded Model Checking
 - ▶ Encode the semantics into a Satisfiability Modulo Theory problem

SMT query = “Is there a trace with correct semantics ?”
assert(\wedge expr)

- SMT-solver response:
 - ▶ SAT: There is a feasible execution path
 - ▶ UNSAT: There is no feasible execution path

- Bounded Model Checking
 - ▶ Encode the semantics into a Satisfiability Modulo Theory problem
- Add execution times on the paths

SMT query = “Is there a trace with correct semantics ?”
assert(\wedge expr)

- SMT-solver response:
 - ▶ SAT:
 - ▶ UNSAT:

- Bounded Model Checking
 - ▶ Encode the semantics into a Satisfiability Modulo Theory problem
- Add execution times on the paths

SMT query = “Is there a trace with correct semantics
assert(\wedge expr)
...such that the execution time is greater than X ?”

- SMT-solver response:
 - ▶ SAT: There is a feasible path with an execution time $> X$
 - ▶ UNSAT: We found a WCET upper bound

- Bounded Model Checking
 - ▶ Encode the semantics into a Satisfiability Modulo Theory problem
- Add execution times on the paths

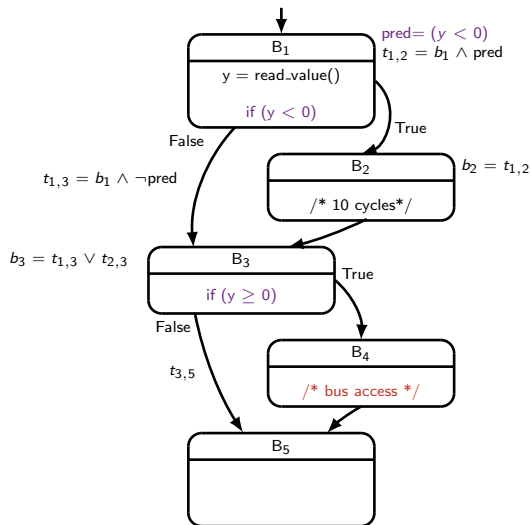
SMT query = “Is there a trace with correct semantics
assert(\wedge expr)
...such that the execution time is greater than X ?”

- SMT-solver response:
 - ▶ SAT: There is a feasible path with an execution time $> X$
 - ▶ UNSAT: We found a WCET upper bound

Goal:

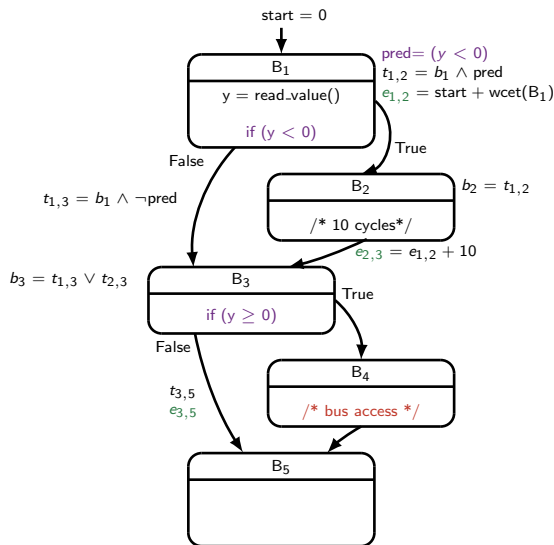
Find the smallest X , such that Execution Time $> X$ is UNSAT

Example: Semantics and Timing Encoding



- ▶ b_i "true" if B_i executed
- ▶ $t_{i,j}$ "true" if transition $B_i \rightarrow B_j$ taken

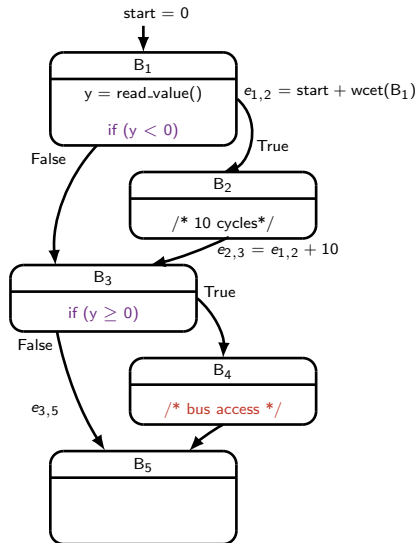
Example: Semantics and Timing Encoding



execution time = if $t_{3,5}$ then $e_{3,5}$ else $e_{4,5}$

- ▶ b_i "true" if B_i executed
- ▶ $t_{i,j}$ "true" if transition $B_i \rightarrow B_j$ taken
- ▶ $e_{i,j}$ execution time at transition $B_i \rightarrow B_j$

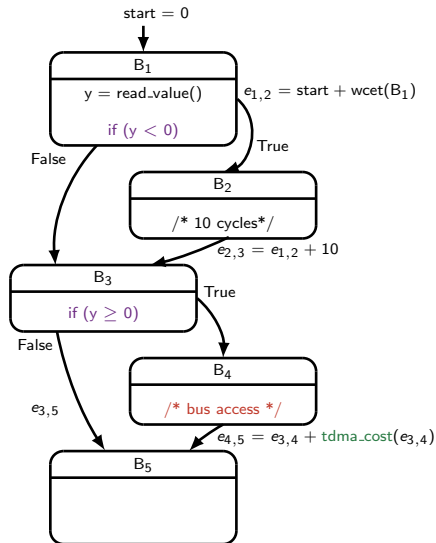
Example: Semantics and Timing Encoding



- $e_{i,j}$ execution time at transition $B_i \rightarrow B_j$

execution time = if $t_{3,5}$ then $e_{3,5}$ else $e_{4,5}$

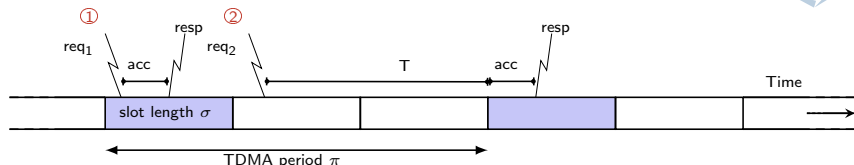
Example: Semantics and Timing Encoding



execution time = if $t_{3,5}$ then $e_{3,5}$ else $e_{4,5}$

- ▶ $e_{i,j}$ execution time at transition $B_i \rightarrow B_j$
- ▶ $\text{tdma_cost}()$ execution time of a bus access

TDMA delay: SMT Encoding

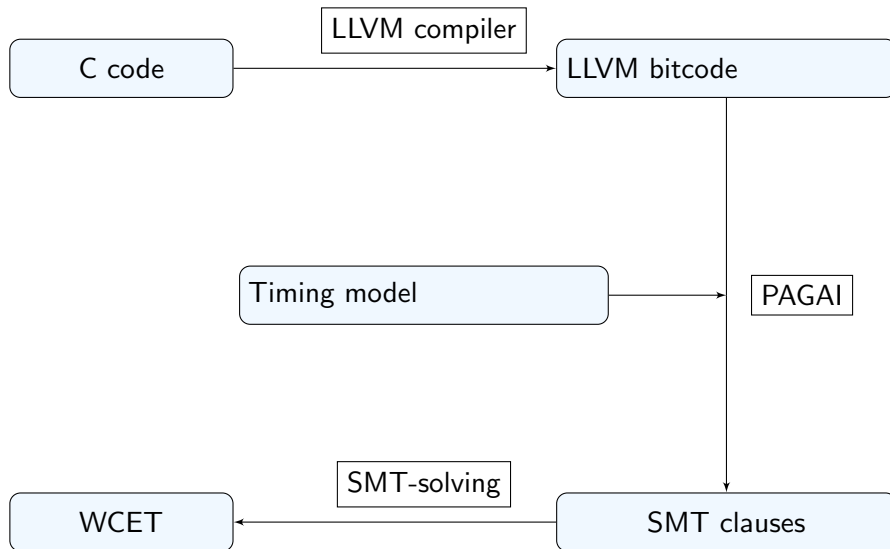


Algorithm *tdma_cost*: returns the execution time of a bus access

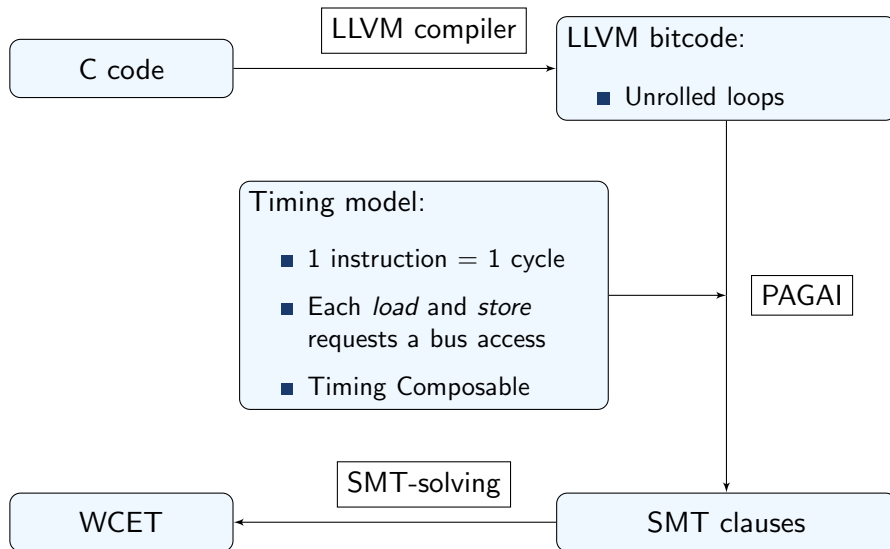
Require: e_{entry}

- 1: $\text{offset}_{\text{entry}} \leftarrow e_{\text{entry}} \bmod \pi$
 - 2: **if** $\text{offset}_{\text{entry}} \in [0, \sigma - \text{acc}[$ **then** /* offset is inside the slot */ ①
 - 3: **return** acc
 - 4: **else** /* offset is outside the slot */ ②
 - 5: **return** $(\pi - \text{offset}_{\text{entry}}) + \text{acc}$
 - 6: **end if**
-

Proof-of-Concept Implementation

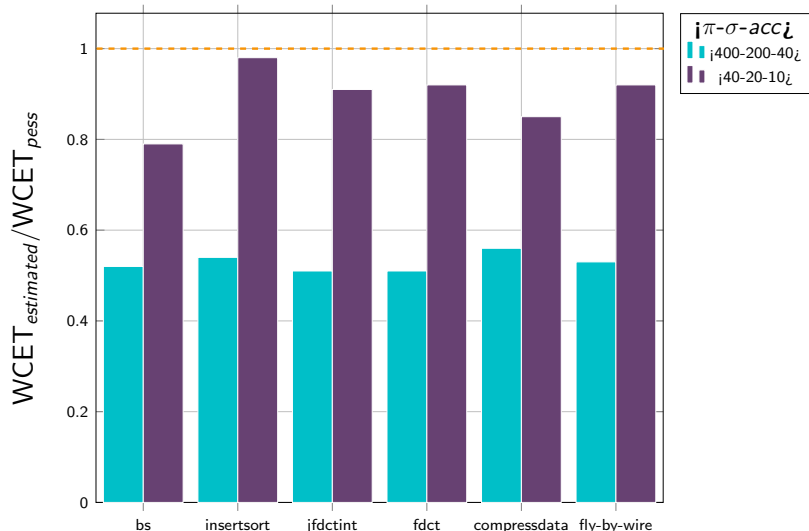


Proof-of-Concept Implementation



Name	Description	#LLVM instr.	#bus access
bs	Binary search	231	12
insertsort	Insertion sort on a reversed array	493	65
jfdctint	Discrete Cosine Transformation	2334	448
fdct	Fast Discrete Cosine Transform	2502	385
compressdata	Data compression program adopted from SPEC95	674	131
fly-by-wire	UAV fly-by-wire software	2815	515

Table: Benchmark description



Estimated WCET of the benchmarks with different configurations of the TDMA bus

Name	$\langle \pi, \sigma, acc \rangle$	
	$\langle 40, 20, 10 \rangle$	$\langle 400, 100, 40 \rangle$
bs	0.45s	0.98s
insertsort	1.37s	6.56s
jfdctint	44.10s	48.54s
fdct	41.36s	34.57s
compressdata	4.66s	3.23s
fly-by-wire	28.78s	149.01s

Analysis time




- SMT encodings for TDMA access
- Validation with small but relevant benchmarks

Why does it work?

- Feasible Path Analysis combined with the WCET computation
- Best-case of gain: All requests are within the TDMA slots
- Worst-case of gain: All requests have worst-case delay

Discussion

- Very good approach for TDMA
- No TT: analysis depends on other tasks = this kind of approach does not scale anymore
- Missing step: implementation on binary level

-  S. Chattopadhyay, A. Roychoudhury, and T. Mitra.
Modeling shared cache and bus in multi-cores for timing analysis.
SCOPES '10, NY, USA.
-  J. Henry, M. Asavoae, D. Monniaux, and C. Maïza.
How to compute worst-case execution time by optimization modulo theory and a clever encoding of program semantics.
LCTES '14, NY, USA.
-  T. Kelter, H. Falk, P. Marwedel, S. Chattopadhyay, and A. Roychoudhury.
Static analysis of multi-core tdma resource arbitration delays.
Real-Time Syst., Mar. 2014.