Worst-Case Energy-Consumption Analysis by Microarchitecture-Aware Timing Analysis for Device-Driven Cyber-Physical Systems

WCET 2019, July 9, 2019

Phillip Raffeck, Christian Eichler, Peter Wägemann, Wolfgang Schröder-Preikschat

Friedrich-Alexander-Universität Erlangen-Nürnberg







FACULTY OF ENGINEERING

Analysis Use Case: Monitoring Bats in the Wild





Motivation: Analysis

Analysis Use Case: Monitoring Bats in the Wild





Monitoring Bats in the Wild

- Communication protocol ~→ time constraints
- Limited energy source ~→ energy constraints

Analysis Use Case: Monitoring Bats in the Wild





Monitoring Bats in the Wild

- Communication protocol ~→ time constraints
- Limited energy source ~> energy constraints
- Guarantee meeting of constraints
- Analysis of worst-case time and energy consumption





Assumptions

- OSEK compliant system
- Full preemption (by interrupts)
- Tasks temporarily activate peripheral devices
- Single-core processor with low complexity
- Timing-anomaly free hardware



















Device-Driven Cyber-Physical Systems





Phillip Raffeck

Motivation: Device-Driven Systems

Influence of Peripheral Devices on WCRE





Influence of Peripheral Devices on WCRE





- Temporarily active devices dominate power consumption
- Affect worst-case response energy consumption (WCRE)



1 ... 2 ldr r1, [r0, #0] 3 ... 5 bx lr 6

















Pessimism Through Missing Microarchitecture Knowledge

- No knowledge of pipeline
 - Assume each instruction executed in isolation
- No knowledge of cache
 - Fetch cost from flash for each instruction
- Overly pessimistic









Phillip Raffeck









- τ_L uses cache blocks 1 and 2 before preemption
- τ_H uses cache blocks 1 and 2
- τ_L uses cache blocks 2 and 3 after preemption





- τ_L uses cache blocks 1 and 2 before preemption
- τ_H uses cache blocks 1 and 2
- τ_L uses cache blocks 2 and 3 after preemption





- τ_L uses cache blocks 1 and 2 before preemption
- τ_H uses cache blocks 1 and 2
- τ_L uses cache blocks 2 and 3 after preemption





- τ_L uses cache blocks 1 and 2 before preemption
- τ_H uses cache blocks 1 and 2
- τ_L uses cache blocks 2 and 3 after preemption

Cache-Related Delays

- τ_H replaces contents of τ_L in cache block 2
- D: reload cache block 2

Influence of Preemption Delays on WCRE





WCRE

- Combine WCRT and power consumption (green area)
- Maximum power consumption
 - Safe bounds

Influence of Preemption Delays on WCRE





WCRE

- Combine WCRT and power consumption (green area)
- Maximum power consumption
 - Safe bounds



Hardware Modeling for Timing Analysis

- Labor-intensive derivation from manual
- Uncertainties in the hardware model
- + Established methods for use in **timing** analysis



Hardware Modeling for Timing Analysis

- Labor-intensive derivation from manual
- Uncertainties in the hardware model
- + Established methods for use in **timing** analysis

- No information on **energetic** behavior
 - Number of transistor switches per instruction?
- Complex data dependencies

Problem Recap and Approach

Problems

- Influence of peripherals on power consumption
- Microarchitecture-aware analysis
 - Microarchitecture state, preemption delays
 - No information about energetic behavior





Problem Recap and Approach

Problems

- Influence of peripherals on power consumption
- Microarchitecture-aware analysis
 - Microarchitecture state, preemption delays
 - No information about energetic behavior

Approach

- Decomposition of the system in all possible **power states**
 - Extend SysWCEC
- Modeling of the microarchitecture state
 - 🖙 Time: direct modeling
 - Energy: indirection over timing analysis
 - Neglect insignificant variances on microarchitecture level
 - Avoid modeling for energetic behavior





Motivation

Background: Microarchitectural Analysis

Microarchitecture-Aware Whole-System Resource Analysis

Evaluation

Conclusion

Microarchitecture Modeling: MEG



Microarchitecture Execution Graph (MEG) [1]

- Directed graph
- Node: microarchitecture state
- Edge: possible transition between states

[1] I. Stein: ILP-based path analysis on abstract pipeline state graphs. Doctoral Thesis. 2010

Microarchitecture Modeling: MEG

Microarchitecture Execution Graph (MEG) [1]

- Directed graph
- Node: microarchitecture state
- Edge: possible transition between states

Microarchitecture State

- Instruction cache state
- Contents and processing time of pipeline stages



[1] I. Stein: ILP-based path analysis on abstract pipeline state graphs. Doctoral Thesis. 2010





Graph Construction

- Known start state for cache and pipeline
- Compute influence of each CPU tick
 - Create node for new microarchitecture state
 - Create transition from predecessor



Graph Construction

- Known start state for cache and pipeline
- Compute influence of each CPU tick
 - Create node for new microarchitecture state
 - Create transition from predecessor

Execution Cost Calculation

Accumulate transition cost

Preemption Delay



Cache-Related Preemption Delay

- Evicting Cache Block (ECB) [2]
- Cache blocks used by preempting task
- Considering only ECB yields competitive results [3]



[2] J. Busquets-Mataix et al.: Adding instruction cache effect to schedulability analysis of preemptive real-time systems. RTAS 1996
[3] D. Shah et al.: Experimental Evaluation of Cache-Related Preemption Delay Aware Timing Analysis. WCET 2018

Preemption Delay



Cache-Related Preemption Delay

- Evicting Cache Block (ECB) [2]
- Cache blocks used by preempting task
- Considering only ECB yields competitive results [3]

Pipeline-Related Preemption Delay [4]

- Preemption and Resume
- Target specific preemption cost

[2] J. Busquets-Mataix et al.: Adding instruction cache effect to schedulability analysis of preemptive real-time systems. RTAS 1996
[3] D. Shah et al.: Experimental Evaluation of Cache-Related Preemption Delay Aware Timing Analysis. WCET 2018

[4] J. Schneider: Cache and Pipeline Sensitive Fixed Priority Scheduling for Preemptive Real-Time Systems. RTSS 2000

au_{H} {1.2} au_{L} {1.2} D {2.3}

Phillip Raffeck

Microarchitecture-Aware Whole-System Resource Analysis

Whole-System WCRE Analysis





Phillip Raffeck

Microarchitecture-Aware Whole-System Resource Analysis





WCRE?

Phillip Raffeck

Microarchitecture-Aware Whole-System Resource Analysis





WCRE?

SysWCEC Approach [5]

- Enumerate operating-system states
- Calculate context-sensitive power consumption per state
- Aggregate basic blocks into atomic regions
 - From a scheduling perspective
 - No system calls during execution
 - Interrupts may occur and release tasks
 - With regard to power-consumption changes \rightsquigarrow Constant set of active devices

[5] P. Wägemann et al.: Whole-System Worst-Case Energy-Consumption Analysis for Energy-Constrained Real-Time Systems. ECRTS 2018

Whole-System WCRE Analysis





Power-State-Transition Graph (PSTG)

- Construct from regions and context knowledge
- Execution paths including system calls and interrupts
- Context-sensitive power consumptions of regions
 - Maximum power consumption per state
 - Variances on microarchitecture level are minor
- Decomposition in power states



init();	1 amily
devOn();	{ 1211100
work();	} o2mW
devOff();	Į
<pre>cleanup();</pre>	} 12mW













WCRE Computation

- Worst-case cost for every PSTG node
 - $\rightsquigarrow~$ Combine region cost and power state
- PSTG \mapsto mathematical optimization problem (ILP)
 - 🖙 objective: WCRE





Power-State-Transition Graph

- Transitions for system calls and interrupts
- Context-sensitive power consumption





Power-State-Transition Graph

- Transitions for system calls and interrupts
- Context-sensitive power consumption
- Utilize context knowledge to bound preemption delays





Power-State-Transition Graph

- Transitions for system calls and interrupts
- Context-sensitive power consumption
- Utilize context knowledge to bound preemption delays
- 🖙 Global power-aware control-flow graph

SysWCEC + Microarchitecture Awareness





Microarchitecture Awareness

- Microarchitecture-aware cost for atomic regions
 - ✓ Microarchitecture Execution Graph
- Microarchitecture awareness for inter-task effects
 - ✓ Preemption delays (pipeline, cache)
 - Include delays in optimization problem

Experimental Setup #1

- Target Platform: Infineon XMC4500
 - ARM Cortex-M4 processor
- Generate benchmarks with known WCET
 - Tools for automation [6, 7]
 - Known baseline
- Compare WCET estimates for benchmarks
 - aiT
 - PLATIN [8]
 - Microarchitecture-aware PLATIN

[6] C. Eichler et al.: Demo Abstract: Tooling Support for Benchmarking Timing Analysis. RTAS 2017

[7] P. Wägemann et al.: Benchmark Generation for Timing Analysis. RTAS 2017

[8] S. Hepp et al.: The Platin Tool Kit - The T-CREST Approach for Compiler and WCET Integration. KPS 2015









Benchmark Number





Evaluation





Evaluation





Benchmark Number

Evaluation





Phillip Raffeck

Evaluation

Experimental Setup #2

- Target Platform: Infineon XMC4500
 - ARM Cortex-M4 processor
- Generate benchmarks with known WCRE
 - Tools for automation [8]
 - Known baseline
- Compare WCRE estimates for benchmarks
 - Pessimistic SysWCEC
 - Microarchitecture-aware SysWCEC

[8] C. Eichler et al.: GenEE: A Benchmark Generator for Static Analysis Tools of Energy-Constrained Cyber-Physical Systems. CPS-IoTBench2019



















Tasksets with Peripherals – Analysis Time





Phillip Raffeck

Conclusion





- Missing microarchitecture knowledge for energy
- 🖙 Exploit microarchitecture awareness in timing analysis
- WCRE analysis via timing analysis



gitlab.cs.fau.de/ svswcec-uarch

- Microarchitecture-aware
- 🗸 System-aware
- 🗸 Device-aware
- Time and energy





- Missing microarchitecture knowledge for energy
- 🖙 Exploit microarchitecture awareness in timing analysis
- WCRE analysis via timing analysis



gitlab.cs.fau.de/ svswcec-uarch

- Microarchitecture-aware
- 🗸 System-aware
- Device-aware
- Time and energy

Questions?

Conclusion