UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

HALMSTAD UNIVERSITY

A Practical Study on WCET Estimation on Multicore Processors for Avionics Applications

Prof. Edison Pignaton de Freitas

B. B. Cozer, C. G. Ferreira, F. R. Wagner and T. Larsson

.inf INSTITUTO DE INFORMÁTICA UFRGS

Agenda

- Introduction
- Background
- Related Works
- Tools and Methods
- Experiments and Results
- Conclusion and Future Work

Introduction

Introduction

Integrated Modular Avionics (IMA)

- Evolution in the early 2000’s
- Integrated software modules running in the same hardware
- RTCA/DO-297 Standard

Avionics Applications

- As safe-critical software avionics need to have their expected behavior known – WCET
- Independance and non-interference
  - Spatial and temporal separation
  - Spatial separation → ARINC653 standard
  - Temporal is harder to analyse

Introduction

Multi-core Processors Evolution

- Scalability and power efficiency for general purpose software execution
- Avionics systems have strict timing concerns!

Goal

Discuss alternatives to improve avionics software WCET analysis to use multi-core processors

Background

Background

WCET Analysis for Multicore processors

- WCET – Upper bound on execution time

- It is a fundamental metric for the development and the validation process of safety-critical systems

- Most techniques involve **static analysis** and/or **measurements** analysis only

- Multicore processors introduce new challenges
  - ➢ Shared internal processor resources (caches, memories and intra-processor communication buses)

Background

WCET Analysis for Multicore processors

- Approaches to deal with the problems and the drawbacks related

Approach	**Drawback**
Shared resources serialization through TDMA schemes	Inefficient resource utilization due to **resource privatization**
Customizations in processor hardware architecture	**Preventing the usage of COTS** processors
Shared Resources Joint Analysis	**Hard scalability** when using multiple cores and **difficult** usage of **incremental development** and **certification**
Response Time Analysis and resource conflicts delays	**Does not consider static scheduling** and **difficult** usage of **incremental development** and **certification**
Monitoring Mechanisms using processor counters	Only monitor the **impact of non-real time tasks** while **does not guarantee** achievement of **hard real time tasks** deadlines.

Background

Avionics Certification

- DO178-B/C (ED12B/C) – Software Considerations in Airbone Systems and Equipment Certification

- Introduction of DALs (Design Assurance Level) – based on a safety assessment process and hazard analysis
  - ➢ Catogorized by the failure type: Catastrophic, Hazardous, Major, Minor and No effect

Failure Condition	DAL	Maximum Failure Rate per Flight Hour
Catastrophic	A	1.0E-9
Hazardous	B	1.0E-7
Major	C	1.0E-5
Minor	D	1.0E-3
No Effect	E	--

Background

Avionics Certification

- Software developement must produce artifacts defined by DO-178 standard
  - SRS, SDD, SDT, STR etc..

- Improvement about how WCET is addressed according to the standard to face the introduction of new technology
  - Review and analysis of source code were no longer enough
  - Compiler, linker and hardware needed to be assessed in the calculation

Related Works

Related Works

Monitoring and WCET Analysis in COTS Multi-core-SoC-based Mixed-Criticality Systems – Jan Nowotsch

- Approach to **determine maximum inter-process interference** when shared resources are used
- Allows independent analysis of application, highly necessary in the avionics industry.
  - ➢ Enables incremental development and certification
  - ➢ Software re-use

Accurate and Efficient Identification of Worst-Case Execution Time for Multicore Processors: **A Survey** – Hamid Mushtaq

- Analysis of WCET for multi-core processors
- **WCET calculation methods**:
  - ➢ Static Analysis
  - ➢ Model Checking
  - ➢ Static + Model
- Highlights as major open issue the data sharing among cores

Related Works

Challenges in Future Avionic Systems on Multi-core Plaftorms – Andreas Lofwenmark

- Enumeration of challenges and open issues for use of multi-core platforms
- Focusing in
  - How to mamage the CPU and memory access
  - How to deal with error corrections and detections on avionics platform – high impact in the access time
  - Access cache memory – private or shared

Chronos: a Timing Analyzer for Embedded Software – Xianfeng Li

- Present Chronos – WCET analysis tool
- Enumeration of COTS solutions for WCET calculation

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

HALMSTAD UNIVERSITY

Tools and Methods

.inf INSTITUTO DE INFORMÁTICA UFRGS

Tools and Methods

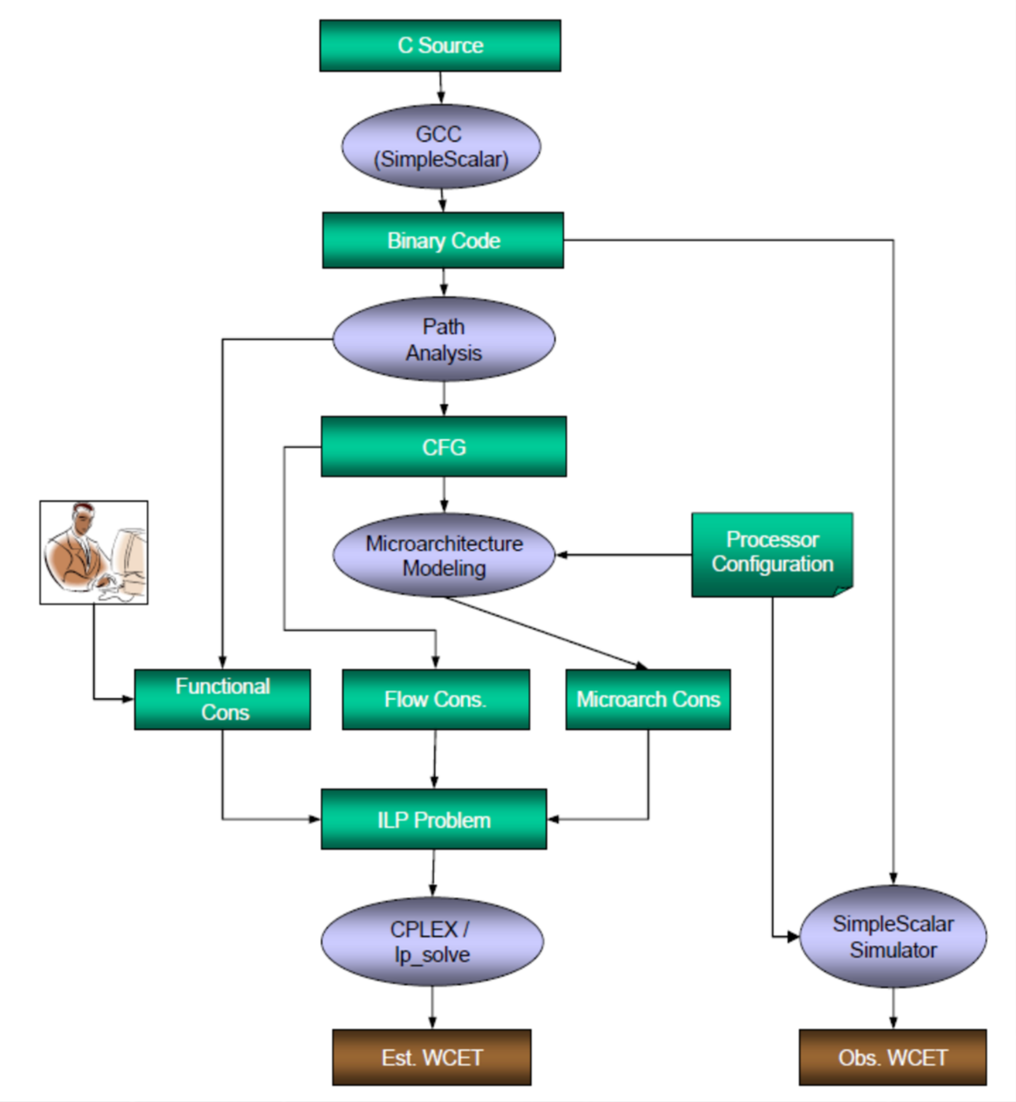
Proposal Overview

- Observation about the need for studies about the usage of multicore platforms for avionics applications
- This work proposes studying alternatives to analyse the impact of multi-core platform in avionics applications
- As starting point, Chronos was selected to study WCET
- Scope: This first step focuses on the private X shared L2 cache memory

Tools and Methods

Chronos – WCET Tool Analyzer

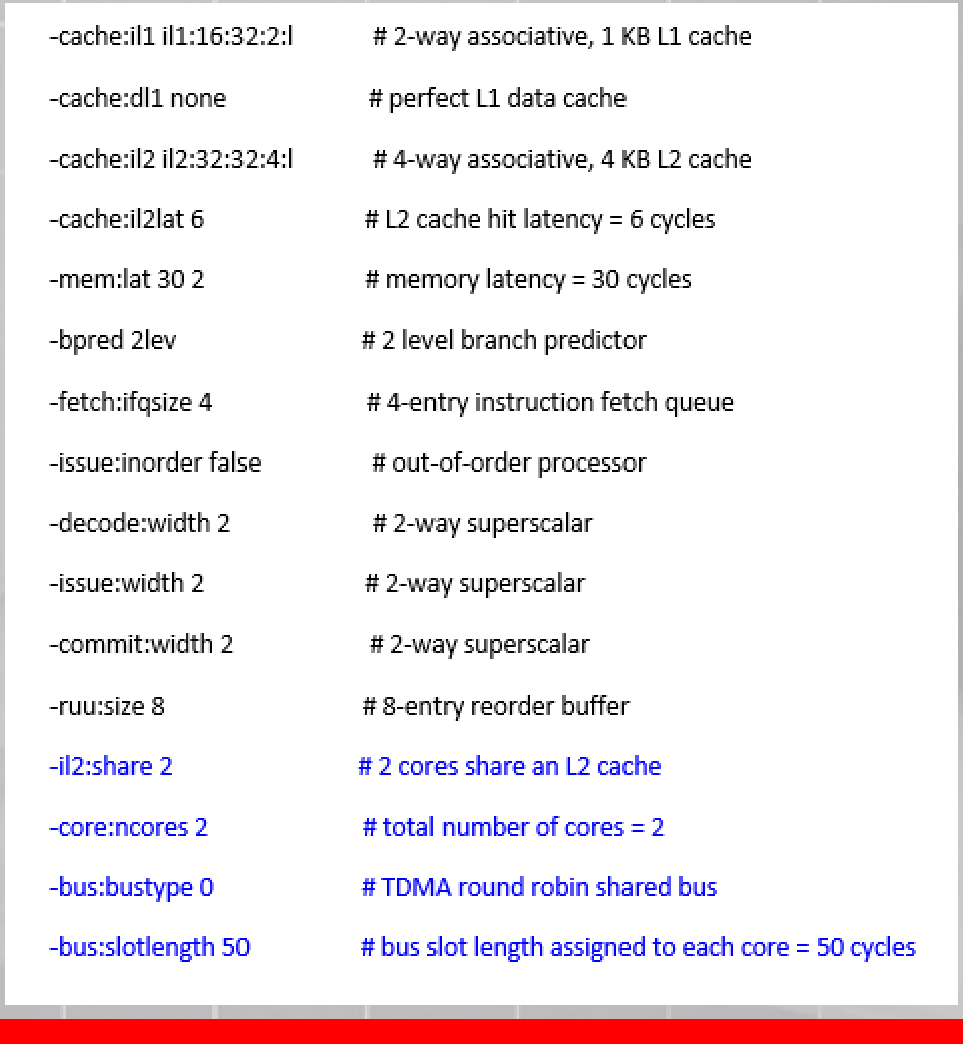
- Static analysis tool that generates WCET estimations
- Involves three sub-tasks: Program path analysis, Microarchitecture modeling and WCET calculation
- Models the processor as in-order and out-of-order pipelines, dynamic branch prediction
- Cache memory is full parametrized
- Introduce concepts of multicore analysis – defining the number of cores, bus allocation and resources sharing



Tools and Methods

Chronos – WCET Tool Analyzer

- Example of processor modeling

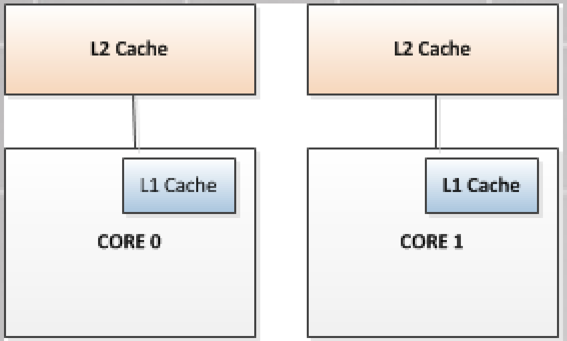


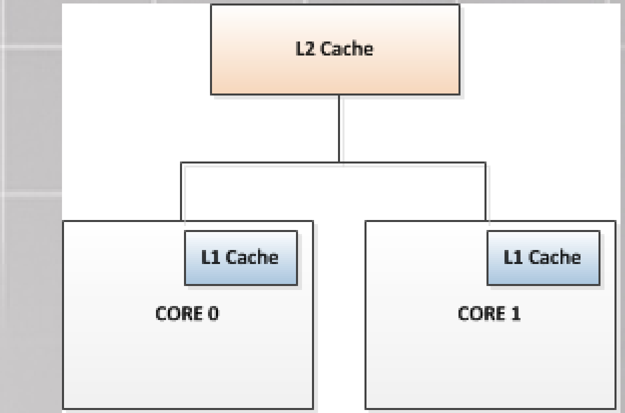
```
-cache:il1 il1:16:32:2:l     # 2-way associative, 1 KB L1 cache
-cache:dl1 none              # perfect L1 data cache
-cache:il2 il2:32:32:4:l     # 4-way associative, 4 KB L2 cache
-cache:il2lat 6              # L2 cache hit latency = 6 cycles
-mem:lat 30 2                # memory latency = 30 cycles
-bpred 2lev                  # 2 level branch predictor
-fetch:ifqsize 4             # 4-entry instruction fetch queue
-issue:inorder false         # out-of-order processor
-decode:width 2              # 2-way superscalar
-issue:width 2               # 2-way superscalar
-commit:width 2              # 2-way superscalar
-ruu:size 8                  # 8-entry reorder buffer
-il2:share 2                 # 2 cores share an L2 cache
-core:ncores 2               # total number of cores = 2
-bus:bustype 0               # TDMA round robin shared bus
-bus:slotlength 50           # bus slot length assigned to each core = 50 cycles
```

Experiments and Results

Experiments

Private x Shared L2 cache





Processor configuration

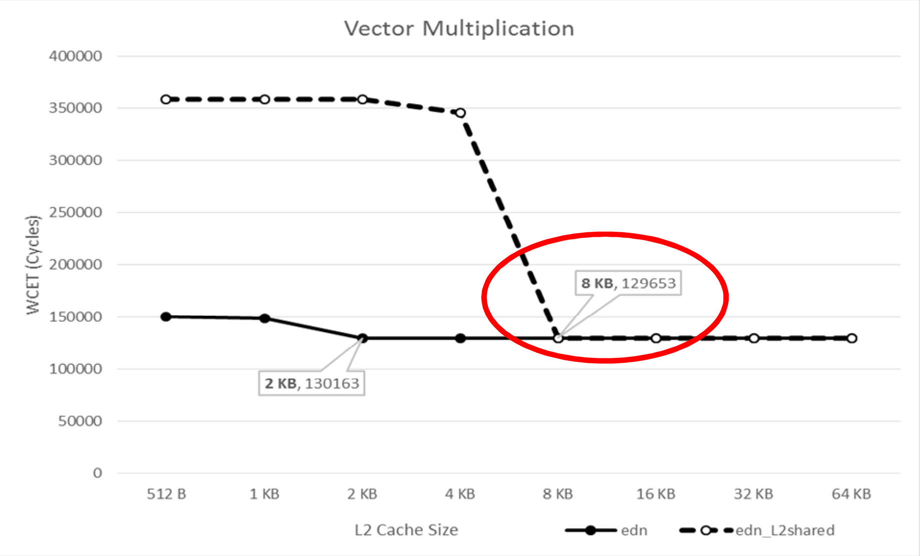
	L1	L2
Scenario 1	128 B	512 B
Scenario 2	128 B	1 KB
Scenario 3	128 B	2 KB
Scenario 4	128 B	4 KB
Scenario 5	128 B	8 KB
Scenario 6	128 B	16 KB
Scenario 7	128 B	32 KB
Scenario 8	128 B	64 KB

Benchmarks

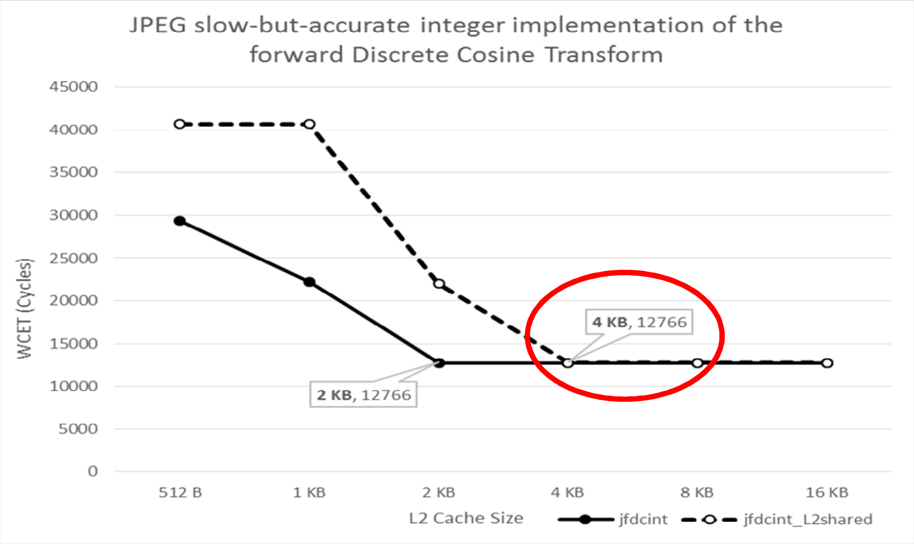
	Meaning	Description
1 – edn	Vector Multiplication	Compilation of several cases which **implements vector multiplications and array handling**.
2 – jfdcint	JPEG slow-but-accurate integer implementation of the forward Discrete Cosine Transform	**Long calculation sequences** (i.e., long basic blocks), **single-nested loops**.
3 – adpcm	Adaptive Differential Pulse Code Modulation algorithm	16Khz sample rate data is used as input data and after calculation the result and **compressed array** are generated.
4 - ndes	Complex embedded code.	A lot of **bit manipulation, shifts, array and matrix calculations**.

Results

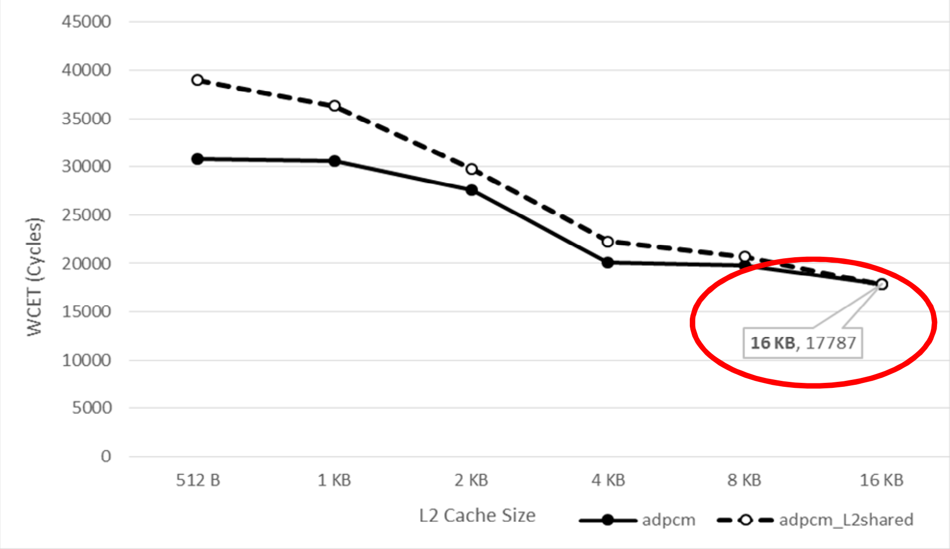
1 - edn benchmark analysis



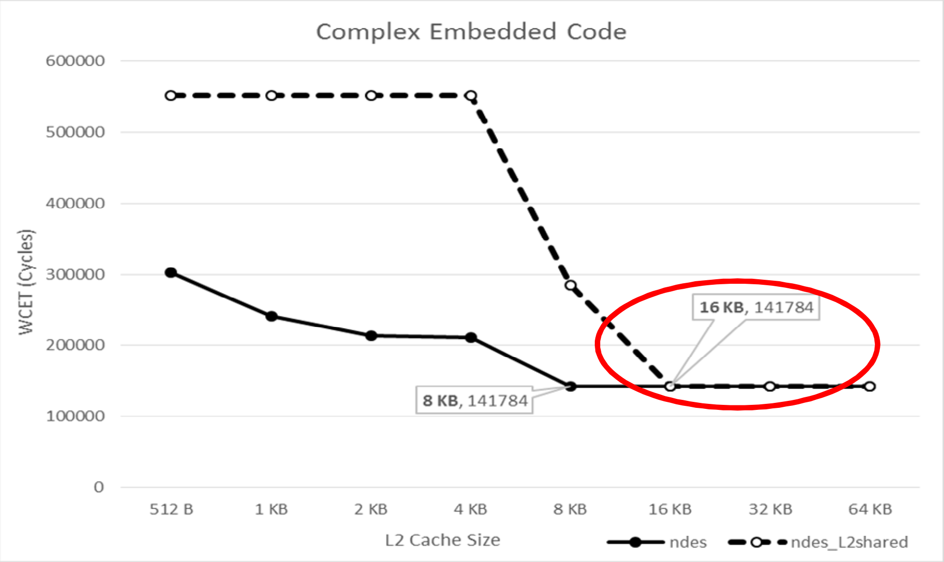
2 - jfdcint benchmark analysis



3 - adpcm benchmark analysis



4 - ndes benchmark analysis



Conclusion and Future Work

Conclusion

- **Shared cache memory causes a major impact over WCET analysis**
  - ➢ Confirming what is described in general by some related works
  - ➢ By consequence, affects the system determinism
  - ➢ Major challenge for multicore system certification

Future Work

- Wide up the analysis on simulated environment using more complex SW
- Execute the same analysis using a real multicore hardware platform
- Expand to processors architecture to model more than 2 cores
  - ➢ Chronos does not support more than two cores

Thank you!

Questions?!

This work is been performend under a cooperation between Federal University of Rio Grande do Sul in Brazil and Halmstad University in Sweden

IMPORTANT NOTICE:

The development and presentation of this work is financially supported by the Swedish-Brazilian Research and Innovation Centre – CISB. The authors thank CISB for the provided support.