Improving Time Predictability of Multicores in Real-time systems: The case of the Space Domain

Javier Jalle
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center

Advisors: Jaume Abella, Eduardo Quiñones, Francisco J. Cazorla
Barcelona Supercomputing Center

Collaborators: Luca Fossati, Marco Zulianello
European Space Agency

javier.jalle@bsc.es
Critical Real-Time Embedded Systems (CRTES)

– Provide evidence about the timing and functional correctness of the system

– **Timing correctness** includes:
  
  • **WCET** (Worst Case Execution Time), Response Time Analysis, Scheduling and Testing
CRTES demand greater performance

**Automotive**
- Advanced driver assistance

**Growing automation**

**Avionics** *source: Airbus*

**Space** *source: Nasa*

Javier Jalle (javier.jalle@bsc.es)
Space Case Study

Active Debris Removal[1]
- Complex autonomous GNC (Guidance and Navigation Control) system and image processing.

What is needed?
- A simple, cost-effective yet high-performance computing solution.
- A test bed environment that allows verification and validation of the complete solution.

Current solution for high performance GNC:
- A single core computer duplicated, triplicated, quadruplicated, …

1 Luisa Innocenti, “User cases: Active Debris Removal”, ADCSS 2013.
Multicores for Hard Real-time systems

Pros:
- Better performance per watt than single-core processors
- Maintain simple core design
- Enable co-hosting mixed-criticality applications
  - Hardware utilization is maximized, while cost, size, weight and power requirements are reduced.
- ...
Multicores for Hard Real-time systems

Cons:

- Require functional isolation
  - Prevent that one application corrupts the state of other applications;
  - Low-criticality applications must not affect high-criticality ones

- Harder to time analyze w.r.t. single-core chips
  - It is hard to provide a safe and tight worst-case execution time (WCET) estimation in multi-cores

- …
Multicores for Hard Real-time systems

**Inter-task interferences:**
- Tasks access hardware resources at the same time.

**Execution time of a task in a multi-core depends on the co-running tasks!**

* Real data obtained from the NGMP-GR740 processor
Architectural solutions for the timing predictability of the Next Generation Multi-Processor (NGMP)

- Ease the adoption of multicore processors by the European Space Agency (ESA) with hardware solutions

- Analyze and improve on-chip shared resources in terms of time predictability and time composability in deterministic and time-probabilistic architectures
Time predictability is the property of a system to derive bounds on the execution time (WCET estimates).

- Tightness of the bounds

Time Composability: *the WCET computed for a task in isolation is not affected by other software in the system*

- Enables incremental qualification and system upgrades

**Time composability is a central element for reducing timing verification and validation costs**
Time predictability and Time Composability

Multicore Architecture

Tasks

Scheduler

WCET

Schedule

Javier Jalle (javier.jalle@bsc.es)
Time predictability and Time Composability

Time Predictable & Time Composable Multicore Architecture

Tasks

Scheduler

WCET

Schedule

Javier Jalle (javier.jalle@bsc.es)
Timing analysis approaches

Deterministic Timing Analysis
- Provides fixed execution times
- Derives WCET estimates
- Requires detailed information from the user

Probabilistic Timing Analysis
- Provides a probability distribution of execution times
- Derives pWCET estimates with a given exceedance probability
- Requires less information from the user
NGMP: On-chip shared resources

**NGMP (GR740)**
- 4 LEON4 cores
- Shared bus
- Partitioned L2
- Shared memory
- Rest of resources are IO, debug and peripherals

*Figure: NGMP block scheme. Source: Cobham Gaisler*

**Hardware is still being improved.**
- Research can influence future developments
Methodology

- Simulation framework
  - Based on SocLib
  - SparcV8 ISA
  - Configurable core (pipeline), bus, cache and memory modules
  - Integration with DRAMsim2

- Real boards
  - GRN2X (used to validate simulation framework)
  - GR740

- Case studies
  - AOCS kernel from EagleEye (European Space Agency)
  - NIR HAWAII On-board Payload (European Space Agency)
  - EEMBC automotive benchmark suite

- Timing analysis
  - RapiTime (Rapita Ltd.)
  - Measurements
Goals of this thesis

- Improve the timing predictability of shared resources in multicore processors
  - Deterministic and probabilistic techniques
  - Time composable WCET estimates
  - Deliver both high average performance and low WCET estimates

- Main focus on on-chip **bus** and **memory controller**
Why buses?

- Real-Time industry has just began to use multicores
  - Low number of cores

- Hierarchical bus configurations scale quite easily to large systems [1]

- Bus-based networks can significantly lower energy consumption and simplify network protocol [2]

The Advanced Microcontroller Bus Architecture (AMBA) is one of the most broadly used bus interfaces.

- Advanced High-performance Bus (AHB)

Figure: AHB architecture
Timing of the Bus

Time to get the bus ownership
Arbitration Policy
Workload = Tasks running on the system

Time to transfer data
Bus properties: width, latency,…

“AMBA does not define the arbitration policy.”
Bus Arbitration policies

Round-robin (RR) and TDMA are most appealing policies for real-time on-chip buses

- Both policies are time composable
- TDMA can offer tighter WCET estimates … if the bus access cycle is known.
- RR offers better average execution time.

<table>
<thead>
<tr>
<th></th>
<th>WCET tightness</th>
<th>Average Perf.</th>
<th>Time Composability</th>
<th>Prioritization</th>
<th>HW/SW changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>RR</td>
<td>✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>TDMA</td>
<td>✓</td>
<td>✗ ✗</td>
<td>✓ ✓</td>
<td>✓ ✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

J. Jalle et al. “Deconstructing bus access control policies for real-time multicores.” SIES 2013
Timing of the Bus

- Both RR and TDMA require a maximum bus latency per request.

- In AMBA the bus latency depends on the particular components in the MPSoC

- NGMP has about 30 different IPs from many providers:
  - Sitael, Cobham Gaisler, Star Dundee, Airbus Space and Defence, Maya Technologies, Arquimea, ICSense, IMEC, …
Time composability:
• Central element for reducing timing verification and validation costs

Bus behavior depends on the specific IPs‘ behavior
– … thus, also depends on the software using them!
– It can even be unbounded!

IPs may be subject to different safety and security levels

Time Composability should be provided by design, by the bus specification itself
Time Composable AMBA AHB

Analyze AMBA features in terms of time-composability
– Time composable restricted version of AMBA => pessimism!

Define a new AMBA specification: AHRB
– AHRB achieves both, time composability and tight WCET estimates

Probabilistic bus designs

PTA relies on execution times modeled by i.i.d random variables

- independent and identically distributed (i.i.d.) random variables:
  - Independence: occurrence of one event doesn’t affect the occurrence of the other event
  - Identical distribution: same probability distribution

Bus latency needs to be an i.i.d random variable
Probabilistic bus designs: Arbitration Policies

- Probability of not being selected after \( k \) rounds
  - Lottery bus
  - Random Permutations
  - Round-Robin (using the worst-case latency)
Probabilistic bus designs

PTA can be used together with multicore processor designs.

- Using Random permutation bus arbitration policy (between 2% and 11% better).
- Hierarchical bus configuration: intra-cluster and inter-cluster bus
- Configurations up to 16 cores.

Why memory controller?

CRTES manage large working sets which requires the use of off-chip memories such as DDR2 and DDR3.

Memory latency has a lot of jitter
- Row-buffer policies
- Refresh operations
- Memory scheduling

Shared resource with the highest impact [1][2]

DRAM Memory architecture

- Memory is composed of different memory banks that can be accessed in parallel
  - Bank-Level Parallelism

- Memory contains a row-buffer from where columns can be read or written.
  - Row-Buffer Locality

- Memory controller serves as interface
  - Row-Buffer Policy
  - Memory-Mapping Scheme
  - Memory Scheduler
High performance and predictability

High-performance and predictability are opposing metrics

Example of the Memory Scheduler:

- FR-FCFS: Fully-exploit row-buffer locality by prioritizing row-hits, but heavily increases access bounds (reordering effect)
- Round-robin: enables deriving tight upper-bound latencies but does not fully exploit row-buffer locality

State of the art techniques optimize of one the two

<table>
<thead>
<tr>
<th>REF</th>
<th>Row Buffer Policy</th>
<th>MMS</th>
<th>Scheduler</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODES2007</td>
<td>Close</td>
<td>Interleaved</td>
<td>CCSP</td>
</tr>
<tr>
<td>CODES2013</td>
<td>Close</td>
<td>Interleaved</td>
<td>Reconfig. TDMA</td>
</tr>
<tr>
<td>DATE2013</td>
<td>Conservative Open</td>
<td>Interleaved</td>
<td>TDMA</td>
</tr>
<tr>
<td>ESL2009</td>
<td>Close</td>
<td>Interleaved</td>
<td>Round-Robin</td>
</tr>
<tr>
<td>CODES2011</td>
<td>Close</td>
<td>Private</td>
<td>TDMA</td>
</tr>
<tr>
<td>RTSS2013&amp;ECRTS2014</td>
<td>Open</td>
<td>Private</td>
<td>FIFO</td>
</tr>
<tr>
<td><strong>High Performance</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTAS2014</td>
<td>Open</td>
<td>Private</td>
<td>FR-FCFS</td>
</tr>
<tr>
<td>ISCA2000</td>
<td>Open</td>
<td>Shared</td>
<td>FR-FCFS</td>
</tr>
<tr>
<td>ISCA2008</td>
<td>Open</td>
<td>Shared</td>
<td>PAR-BS</td>
</tr>
</tbody>
</table>
Space Case Study

Dual-criticality in Space applications[1]:

1. Control
   - Predictability
     - Real-time constraints
     - Low memory footprint
     - Nothing indicates that they will be parallelized in the near future

2. Payload
   - High performance
     - Soft- or no-real-time constraints
     - Larger memory footprint
     - Can be parallelized

---

1 ESA contract 4200023100, System Impact of Distributed Multi-core Systems

Javier Jalle (javier.jalle@bsc.es)
Banks are configured as Real-time or High-performance

Tasks are assigned to banks according to their needs
- Private bank: Less interference, may have poor memory utilization
- Shared bank: More interference, better memory utilization
Two-level hierarchical Schedulers

- Intra-bank (within a bank):
  - Real-time banks: Round-robin (bounds the interference)
  - High-performance banks: FR-FCFS (exploits locality and parallelism)

- Inter-bank (between banks):
  - Uses Round-robin (bounds the interference) giving priority for real-time banks. Exploits bank-level parallelism

Control applications are able to obtain tight WCET estimates.

Compared against a FR-FCFS controller.

– We use three workloads: control, balanced and payload based

J. Jalle et al. “A Dual-Criticality Memory Controller (DCmc): Proposal and Evaluation for a Space Case Study..” RTSS 2014
Payload applications do not have a significant performance penalty

- They enjoy a high-performance memory system
- Minimum impact because control applications have low memory usage
Summary & Future work

Improve the timing predictability of shared resources in multicore processors

- Hardware solutions
- Deterministic and probabilistic techniques
- Deliver both high average performance and low WCET estimates
- For on on-chip bus and memory controller
Published work


Collaboration work

We are also researching into software-based solutions
- COTS processors
- No (or limited) hardware support is provided


Improving Time Predictability of Shared Hardware Resources in Real-Time Multicore Systems

Javier Jalle
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center

Advisors: Jaume Abella, Eduardo Quiñones, Francisco J. Cazorla
Barcelona Supercomputing Center

Collaborators: Luca Fossati, Marco Zulianello
European Space Agency

javier.jalle@bsc.es