Evaluation of resource arbitration methods for multi-core real-time systems

Paper presentation at WCET Workshop 2013, Paris

Timon Kelter, Tim Harde, Peter Marwedel

Heiko Falk

Department of Computer Science
TU Dortmund, Germany

Institute of Embedded Systems/Real-Time Systems
Ulm University, Germany
**Predictability for Multicore-Platforms**

Timing influence of parallel task execution

- **Major problem:** Contention on shared resources
  - **Option 1:** Reduce sharing / Duplicate resources
    → Wastes economic potential, some communication is unavoidable
  - **Option 2:** Provide deterministic and analyzable arbitration
    → Needs new analysis methods

→ Local bounds for arbitration delay of individual accesses
Predictability for Multicore-Platforms

Timing influence of parallel task execution

- Major problem: Contention on shared resources
  - Option 1: Reduce sharing / Duplicate resources
    → Wastes economic potential, some communication is unavoidable
  - Option 2: Provide deterministic and analyzable arbitration
    → Needs new analysis methods

→ Local bounds for arbitration delay of individual accesses
Outline

1) System model
2) Arbitration methods
3) Analysis framework
4) Benchmark Setup
5) Evaluation
6) Summary
System model

Core 1
- ARM7TDMI Core
  - D-SPM
  - I-Cache
  - D-Cache
  - Bridge
  - I-SPM

Core N
- ARM7TDMI Core
  - D-SPM
  - I-Cache
  - D-Cache
  - Bridge

Shared bus with configurable arbitration

I-RAM (Uncached)
- L2 I-Cache
- I-RAM (Cached)

D-RAM (Uncached)
- L2 D-Cache
- D-RAM (Cached)

BootROM

Implemented in CoMET/Virtualizer [8]

→ Flexible experimentation platform
Bus arbitration methods

- "Classic" methods (Utilization)
  - Fixed Priority (PRIO)
    - Priority value for each core
    - Non-preemptable access
  - Fair (Round-Robin) (FAIR)

- Time-triggered methods (Predictability)
  - Time-Division Multiple Access (TDMA)
  - Priority Division (PD)
    - Slots of length \( n \) for each slot
    - Priorities \( p_{ij} \) for core \( i \) in slot \( j \)
      - \( p_{11} = \text{max} \)
      - \( p_{22} = \text{max} \)
      - \( p_{33} = \text{max} \)
      - \( p_{44} = \text{max} \)

→ Comparison of achieveable
  - WCET
  - ACET
  - Bus Utilization
Bus arbitration methods

- „Classic“ methods (Utilization)
  - **Fixed Priority** (PRIO)
    Priority value $p_i$ for each core $i$ (non-preemtable access)
  - **Fair** (Round-Robin) (FAIR)

- Time-triggered methods (Predictability)
  - **Time-Division Multiple Access** (TDMA)
    - $n$ Slots of length $l$, owner core $o_j$ for each slot $j$
    
    $$
    \begin{array}{l}
    o_1 = 1 \\
    o_2 = 2 \\
    o_3 = 3 \\
    o_4 = 4
    \end{array}
    $$
  
    - **Priority Division** (PD)
    - $n$ Slots of length $l$, priorities $p_{ij}$ for core $i$ in slot $j$
    
    $$
    \begin{array}{l}
    p_{11} = \text{max} \\
    p_{22} = \text{max} \\
    p_{33} = \text{max} \\
    p_{44} = \text{max}
    \end{array}
    $$
Memory hierarchy analysis options

- Employed approach: Generalized combined analysis ([4], aiT)

- Per-core CFG-based data flow analysis
- Memory accesses are handled by hierarchical state update
- Each stage may forward or handle (e.g. guaranteed cache hit)
- Timing information is exchanged along with general access information
Shared Bus Analysis

- What is the “state” for the shared bus?
  → Approximation of the current position in the cyclic schedule
- Position: Offset from the beginning of the last TDMA period

<table>
<thead>
<tr>
<th>Time</th>
<th>Offsets</th>
<th>Core 1 Slot</th>
<th>Core 2 Slot</th>
<th>Core 3 Slot</th>
<th>Core 4 Slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>x + 1 \cdot l</td>
<td>x + 2 \cdot l</td>
<td>x + 3 \cdot l</td>
<td>x + 4 \cdot l</td>
</tr>
</tbody>
</table>

Abstract Bus State

- Abstraction: Set of offsets $O_b^{\text{in}} \subseteq \{0, \ldots, n \cdot l - 1\}$ transfer $O_b^{\text{out}}$
Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access $a_i$ with spent time $T_{a_i}$ since $a_{i-1}$
  - Forwarding to next stages yields post-bus runtime $D$

$$O^{i+1}_b = \bigcup_{o \in O^i_b, t \in T_{a_i}} \{ \Phi_c(o + t mod n \cdot l) \} \oplus D$$

$$\Phi^{TDMA}_c(o) = \begin{cases} \{ o \} & \text{if } o \in \omega_{\text{must}} \text{ grant immediately} \\ \lfloor \omega_{\text{must}} \rfloor & \text{else} \end{cases}$$

$$\Phi^{PD}_c(o) = \begin{cases} \{ o \} \oplus \{ 0, \ldots, m_{\text{max}} - 1 \} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \lfloor \omega_{\text{must}} \rfloor & \text{if } \exists \omega_{\text{must}} \\ \emptyset & \text{else} \end{cases}$$
Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access $a_i$ with spent time $T_{a_i}$ since $a_{i-1}$
  - Forwarding to next stages yields post-bus runtime $D$

\[
O_{b}^{i+1} = \bigcup_{o \in O_b^i, t \in T_{a_i}} \{ \Phi_c(o + t \mod n \cdot l) \} \oplus D
\]

\[
\Phi_{c}^{TDMA}(o) = \begin{cases} 
\{ o \} & \text{if } o \in \omega_{\text{must}} \\
\lfloor \omega_{\text{must}} \rfloor & \text{else}
\end{cases}
\]

\[
\Phi_{c}^{PD}(o) = \begin{cases} 
\{ o \} \oplus \{ 0, \ldots, m_{\text{max}} - 1 \} & \text{if } o \in \omega_{\text{must}} \\
\phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \lfloor \omega_{\text{must}} \rfloor & \text{if } \exists \omega_{\text{must}} \\
\emptyset & \text{else}
\end{cases}
\]
Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access $a_i$ with spent time $T_{a_i}$ since $a_{i-1}$
  - Forwarding to next stages yields post-bus runtime $D$

$$O_b^{i+1} = \bigcup_{o \in O_b^i, t \in T_{a_i}} \left\{ \Phi_c(o + t \mod n \cdot l) \right\} \oplus D$$

$$\Phi^\text{TDMA}_c(o) = \begin{cases} \{o\} & \text{if } o \in \omega_{\text{must}} \\ \{\floor{\omega_{\text{must}}}\} & \text{else} \end{cases}$$

$$\Phi^\text{PD}_c(o) = \begin{cases} \{o\} \oplus \{0, \ldots, m_{\text{max}} - 1\} & \text{if } o \in \omega_{\text{must}} \\ \phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \{\floor{\omega_{\text{must}}}\} & \text{if } \exists \omega_{\text{must}} \\ \emptyset & \text{else} \end{cases}$$

Grant, with possible lower prio access
Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access $a_i$ with spent time $T_{a_i}$ since $a_{i-1}$
  - Forwarding to next stages yields post-bus runtime $D$

\[
O^{i+1}_b = \bigcup_{o \in O^i_b, t \in T_{a_i}} \{ \Phi_c(o + t \mod n \cdot l) \} \oplus D
\]

\[
\Phi^\text{TDMA}_c(o) = \begin{cases} 
\{ o \} & \text{if } o \in \omega^\text{must} \\
\lfloor \omega^\text{must} \rfloor & \text{else}
\end{cases}
\]

\[
\Phi^\text{PD}_c(o) = \begin{cases} 
\{ o \} \oplus \{ 0, \ldots, m_{\text{max}} - 1 \} & \text{if } o \in \omega^\text{must} \\
\phi_c(\omega(o) \rightarrow \omega^\text{must}) \cup \lfloor \omega^\text{must} \rfloor & \exists \omega^\text{must} \\
\emptyset & \text{else}
\end{cases}
\]

Wait for “own” slot, collect “may”-slot offsets
Shared Bus Analysis (TDMA & PD)

- Transfer function for the shared bus state?
  - Pipeline analysis passes in access \( a_i \) with spent time \( T_{a_i} \) since \( a_{i-1} \)
  - Forwarding to next stages yields post-bus runtime \( D \)

\[
O_{b}^{i+1} = \bigcup_{o \in O_{b}^i, t \in T_{a_i}} \{ \Phi_c(o + t \mod n \cdot l) \} \oplus D
\]

\[
\Phi_{c_{TDMA}}(o) = \begin{cases} 
\{ o \} & \text{if } o \in \omega_{\text{must}} \\
\left\lfloor \omega_{\text{must}} \right\rfloor & \text{else}
\end{cases}
\]

\[
\Phi_{c_{PD}}(o) = \begin{cases} 
\{ o \} \oplus \{ 0, \ldots, m_{\text{max}} - 1 \} & \text{if } o \in \omega_{\text{must}} \\
\phi_c(\omega(o) \rightarrow \omega_{\text{must}}) \cup \left\lfloor \omega_{\text{must}} \right\rfloor & \text{if } \exists \omega_{\text{must}} \\
\emptyset & \text{else}
\end{cases}
\]

No “own” slot exists \( \rightarrow \) Not boundable
Pessimistic Analyses (PRIO & FAIR)

- Local bounds for PRIO & FAIR:
  Need all parallel access interleavings (parallel analysis)

→ Revert to worst-case assumptions in per-core analysis

\[
\Phi_{c}^{\text{PRIO}}(o) = \begin{cases} 
\{o\} \oplus \{0, \ldots, m_{\max} - 1\} & \text{if } c \text{ is max prio core} \\
\emptyset & \text{else}
\end{cases}
\]

\[
\Phi_{c}^{\text{FAIR}}(o) = \{o\} \oplus \{0, \ldots, (n-1) \cdot m_{\max} - 1\}
\]

- Arbitration delay bound function analogous to \(\Phi_{c}\)

→ Transfer & Meet (Set union) functions for DFA

Analogous to PD cases
Pessimistic Analyses (PRIO & FAIR)

- Local bounds for PRIO & FAIR:
  Need all parallel access interleavings (parallel analysis)
  → Revert to worst-case assumptions in per-core analysis

\[
\Phi_c^{\text{PRIO}}(o) = \begin{cases} 
\{o\} \oplus \{0, \ldots, m_{\text{max}} - 1\} & \text{if } c \text{ is max prio core} \\
\emptyset & \text{else}
\end{cases}
\]

\[
\Phi_c^{\text{FAIR}}(o) = \{o\} \oplus \{0, \ldots, (n-1) \cdot m_{\text{max}} - 1\}
\]

- Arbitration delay bound function analogous to \(\Phi_c\)
  → Transfer & Meet (Set union) functions for DFA

Single access from every other core at max
Benchmarking Method

- Standard multicore benchmarks (SPEC, EEMBC, BDTI):
  - Unpredictable behavior of req. OS/middleware software stack
- Aggregate known single-thread benchmarks (MRTC / UTDSP / MiBench / MediaBench / DSPStone, 110 benchmarks in total)
  - Allocate single-thread task to each single-thread core

- How to form balanced task sets?

- Parametrization:
  - Minimal slot length $l = m_{\text{max}}$
  - Memory access times: 1 cycle (L1), 3 cycles (L2)
  - Map (only) global variables to Shared Memory (→ IO-Devices)
WCET Evaluation (Maximum Overestimation)

The chart illustrates the WCET evaluation for different numbers of cores (x1, x2, x4, x8). The overestimation for each method is presented as a percentage of the actual WCET.

- **FAIR**
- **PD (half TDMA)**
- **PD (linear)**
- **TDMA**

**Linear increase due to worst-case assumption**

**Higher overestimation due to accesses in other cores' slots**

Kelter, Harde, Marwedel and Falk: “Evaluation of resource arbitration methods [...]”
ACET Evaluation (Baseline: 1-Core, FAIR)

- Extremely low overhead for FAIR / PRIO
- Inacceptable overhead for rising core numbers
- Scales better than TDMA

Average Relative ACET

Number of cores: x2, x4, x8
Total Bus Utilization Results

FAIR/PRIO: Almost linear scaling

Less steep increase for PD

TDMA: Approximatively constant!
Summary / Future Work

- Combined state-based analysis framework for shared resources
- Evaluation of arbitration policies for a configurable multi-core ARM platform
  - TDMA incurs serious ACET overhead with rising core count
  - PD can balance WCET, ACET and resource utilization
  - FAIR/PRIO provide unmatched utilization

- Extensions:
  - Optimization of TDMA / PD schedules
  - Extension of state-based approach to true parallel analysis
  - Analysis of dependent / cooperative threads
References

References


