On Spin Locks in AUTOSAR: Blocking Analysis of FIFO, Unordered, and Priority-Ordered Spin Locks

Alexander Wieder and Björn Brandenburg
MPI-SWS

RTSS 2013
12/04/2013
Vancouver, Canada
Motivation:

AUTOSAR: OS-specification widely used for embedded applications

Resources accessed from multiple cores:

AUTOSAR mandates spin locks.
Motivation:

AUTOSAR: OS-specification widely used for embedded applications

Resources accessed from multiple cores:
AUTOSAR mandates spin locks.

But AUTOSAR does not specify spin lock type!
Motivation:

AUTOSAR: OS-specification widely used for embedded applications

Resources accessed from multiple cores: AUTOSAR mandates spin locks.

But AUTOSAR does not specify spin lock type!

Which type should be used?
Spin Lock Types

Variety of reasonable choices:

- Non-Preemptable Spinning
- Preemptable Spinning
- FIFO-ordered (MSRP)
- Priority-Ordered
- Unordered
- Priority-Ordered with FIFO tie-breaking
Spin Lock Types

Variety of reasonable choices:

- analysis-friendly
- support for locking priorities
- simple implementation (test&set)
- low architecture requirements
- safe for unknown lock types
- FIFO-ordered (MSRP)
- Priority-Ordered
- Unordered
- Priority-Ordered with FIFO tie-breaking
Spin Lock Types

Variety of reasonable choices:

- Priority-Ordered
- Priority-Ordered with FIFO tie-breaking
- Unordered
- FIFO-ordered (MSRP)
- Non-Preemptable
- Preemptable
- Spinning

No blocking analysis available for most types!
Contributions

Blocking analysis for 8 types of spin locks
Contributions

Blocking analysis for 8 types of spin locks

7/8 spin lock types: first blocking analysis
Contributions

Blocking analysis for 8 types of spin locks

7/8 spin lock types: first blocking analysis

Asymptotically less pessimistic than prior approaches
Contributions

Blocking analysis for 8 types of spin locks

7/8 spin lock types: first blocking analysis

Asymptotically less pessimistic than prior approaches

Suggest AUTOSAR API changes based on evaluation results
Novel Spin Lock Analysis

Key Technique:

Blocking Analysis modeled as Integer Linear Program (ILP)
Novel Spin Lock Analysis

Key Technique:

Blocking Analysis modeled as Integer Linear Program (ILP)

Worst-case blocking bounds determined by ILP solver
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.
Task Model

- sporadic tasks: \( T_i : (e_i, d_i, p_i) \)

- constrained deadlines: \( d_i \leq p_i \)

- partitioned fixed-priority scheduling
Spin locks **busy-wait** while waiting for contended resource.
Basic Spin Lock Analysis

Spin locks **busy-wait** while waiting for contended resource.

Straight-forward analysis approach:

- treat spin-time as execution time
- apply response-time analysis
FIFO-Ordered Spin Locks

Multiprocessor Stack Resource Policy (MSRP) [1]

FIFO-Ordered Spin Locks

Multiprocessor Stack Resource Policy (MSRP) [1]

The MSRP uses non-preemptable FIFO-ordered spin locks for resources shared across processors.

FIFO-Ordered Spin Locks

classic MSRP analysis

\[ \tau \]

inflate execution time

\[ \tau' \]

response-time analysis

\[ \tau' \]
FIFO-Ordered Spin Locks

classic MSRP analysis

inflate execution time

response-time analysis

\( \tau \)

\( \tau' \)

\( \tau' \)

inflated execution times
FIFO-Ordered Spin Locks

classic MSRP analysis

inflate execution time

response-time analysis

\[ R_i = e_i' + \sum_{h<i} \left[ \frac{R_i}{p_h} \right] \cdot e_h' \]
FIFO-Ordered Spin Locks

classic MSRP analysis

\[ R_i = e_i' + \sum_{h<i} \left[ \frac{R_i}{p_h} \right] \cdot e_h' \]
Execution-Time Inflation in Classic MSRP Analysis

- **T1**: Executing
- **T2**: Spinning
- **P1**: T1
- **T2**: Pending but not scheduled
- **P2**: T3
- **T3**: Critical Section

Diagram showing the execution of threads and their states over time (t).
Execution-Time Inflation in Classic MSRP Analysis

- **T1**: Busy-waiting for shared resource
- **T2**: Spinning
- **T3**: Executing
- **P1**
- **P2**

**Critical Section**
Execution-Time Inflation in Classic MSRP Analysis

T2’s job pending, but not executing

Executing

Critical Section

P1

T1

T2

P2

T3

Executing

Pending but not scheduled
Execution-Time Inflation in Classic MSRP Analysis

T1

P1

T1

T2

P2

T2’s job completes

T3

Executing

Spinning

scheduled

Section

T2’s job completes
Pending but not scheduled

Consider T1’s request

Execution-Time Inflation in Classic MSRP Analysis

- Executing
- Spinning
- Pending but not scheduled
- Critical Section
Execution-Time Inflation in Classic MSRP Analysis

Consider T1’s request could be blocked by T3’s request.
Execution-Time Inflation in Classic MSRP Analysis
Execution-Time Inflation in Classic MSRP Analysis

Execution time inflation:
Spin time treated as execution time
Execution-Time Inflation in Classic MSRP Analysis

Execution time inflation:
Spin time treated as execution time
Execution-Time Inflation in Classic MSRP Analysis
Execution-Time Inflation in Classic MSRP Analysis

- **Pending but not scheduled**
- **Executing**
- **Spinning**
- **Critical Section**

**original execution time**

**inflated execution time**
Execution-Time Inflation in Classic MSRP Analysis

- T1 (Executing)
- T3 (Spinning)

Execution time inflated for all jobs!
Execution-Time Inflation in Classic MSRP Analysis

Execution time inflated for all jobs!
Original Schedule

P1

T1

P2

T2

T3

Executing

Spinning

scheduled

Critical Section

T2’s job completes
Schedule with Inflated Execution Times

Pending but not scheduled

Executing

Critical Section

Spinning

T1

T2

T3

P1

P2

Deadline Miss

Executing

Spinning
Schedule with Inflated Execution Times

Impossible in a real schedule!

Deadline Miss

Pending but not scheduled

Executing

Critical Section

Spinning

Schedule with Inflated Execution Times

T1

T2

T3

P1

P2
At most one can be blocked!
Inflation is Inherently Pessimistic

All prior analyses rely on execution time inflation!
Inflation is Inherently Pessimistic

All prior analyses rely on execution time inflation!

We show that execution time inflation is an inherent source of pessimism in blocking analysis.
Inflation is Inherently Pessimistic

Theorem

Any blocking analysis relying on the inflation of job execution costs can be pessimistic by a factor of $\Omega(\phi \cdot n)$. 
Inflation is Inherently Pessimistic

**Theorem**

Any blocking analysis relying on the inflation of job execution costs can be pessimistic by a factor of $\Omega(\phi \cdot n)$.

- maximal ratio of shortest and longest task period
- number of tasks
Inflation is Inherently Pessimistic

Theorem
Any blocking analysis relying on the inflation of job execution costs can be pessimistic by a factor of $\Omega(\phi \cdot n)$.

Details and proof in paper
Challenges

Prior analysis is **pessimistic** due to inflation.

Prior analysis is **specific** to non-preemptable FIFO-ordered spin locks.
Challenges

Prior analysis is **pessimistic** due to inflation.

Prior analysis is **specific** to non-preemptable FIFO-ordered spin locks.

ILP formulation
Explicit blocking terms
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

ILP formulation
Explicit blocking terms
ILP-Based Blocking Analysis of Spin Locks

- inflation-free blocking analysis
- response-time analysis with uninflated execution times
ILP-Based Blocking Analysis of Spin Locks

Key technique for inflation-free blocking analysis:

Blocking Analysis modeled as \textbf{Integer Linear Program} (ILP)

Worst-case blocking bounds determined by ILP solver
ILP Generation for FIFO-Ordered Spin Locks

P1 [T1]

P2 [T2]

P3 [T3]

 Executing  Spinning  Critical Section
ILP Generation for FIFO-Ordered Spin Locks

- P1, T1: Executing
- P2, T2: Spinning
- P3, T3: Critical Section

$t$
Assign blocking variables:
ILP Generation for FIFO-Ordered Spin Locks

Assign *blocking variables*:

- **P1**
  - **T1**
  - \(X_{2,1,1}\)
- **P2**
  - **T2**
  - \(X_{2,1,2}\)
  - \(X_{2,1,3}\)
- **P3**
  - **T3**
  - \(X_{3,1,1}\)
  - \(X_{3,1,2}\)

Legend:
- **Executing**
- **Spinning**
- **Critical Section**
Blocking Variables

\[ 0 \leq X \leq 1 \]

\[ X : \] Fraction of critical section length contributing to T1’s blocking
Blocking Variables

\[ 0 \leq X \leq 1 \]

\( X \):  
Fraction of critical section length contributing to T1’s blocking

\( X = 0 \)
Request \textbf{does not contribute} to T1’s blocking.

\( X = 1 \)
Request \textbf{contributes} to T1’s blocking.
Blocking Variables

\[ 0 \leq X \leq 1 \]

- \( X = 0 \): Request does not contribute to T1’s blocking.
- \( X = 1 \): Request contributes to T1’s blocking.

Analysis accounts at most once for each request.
Blocking Variables

\[ 0 \leq X \leq 1 \]

Analysis accounts at most once for each request

No double counting!
Impose constraints:

ILP Generation for FIFO-Ordered Spin Locks

- Executing
- Spinning
- Critical Section
ILP Generation for FIFO-Ordered Spin Locks

ImPOSE constraints:

Observation

Each request can be blocked by at most one request for the same resource from each other processor.
ILP Generation for FIFO-Ordered Spin Locks

Impose constraints:

\[ X_{2,1,1}, X_{2,1,2}, X_{2,1,3}, X_{3,1,1}, X_{3,1,2} \]

- Executing
- Spinning
- Critical Section
ILP Generation for FIFO-Ordered Spin Locks

Impose constraints:

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
ILP Generation for FIFO-Ordered Spin Locks

Impose constraints:

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

constraints to rule out impossible schedules

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
ILP Generation for FIFO-Ordered Spin Locks

Impose constraints:

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

- constraints to rule out impossible schedules

ILP Constraints
- simple
- composable

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
ILP for Worst-Case Blocking

Generate Integer Linear Program:

\[
X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1
\]

\[
X_{3,1,1} + X_{3,1,2} \leq 1
\]
ILP for Worst-Case Blocking

Generate Integer Linear Program:

- worst-case blocking
  \[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

- maximal blocking
ILP for Worst-Case Blocking

Generate Integer Linear Program:

maximize

\[ X_{2,1,1} \cdot L_{2,1,1} + X_{2,1,2} \cdot L_{2,1,2} + X_{2,1,3} \cdot L_{2,1,3} \]
\[ + X_{3,1,1} \cdot L_{3,1,1} + X_{3,1,2} \cdot L_{3,1,2} \]

subject to

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]
\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
ILP for Worst-Case Blocking

Generate Integer Linear Program:

\[
\text{maximize}
\]

\[
X_{2,1,1} \cdot L_{2,1,1} + X_{2,1,2} \cdot L_{2,1,2} + X_{2,1,3} \cdot L_{2,1,3}
+ X_{3,1,1} \cdot L_{3,1,1} + X_{3,1,2} \cdot L_{3,1,2}
\]

subject to

\[
X_{2,1,1} \leq 1
\]

\[
X_{3,1,1} + X_{3,1,2} \leq 1
\]

maximal critical section length
ILP for Worst-Case Blocking

Generate Integer Linear Program:

maximize

subject to

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]

Worst-case blocking bound determined by ILP solver.
Explicit Accounting for Blocking

\[ R_i = e_i + b_i + \sum_{h<i} \left\lfloor \frac{R_i}{p_h} \right\rfloor \cdot e_h \]
Explicit Accounting for Blocking

\[ R_i = e_i + b_i + \sum_{h<i} \left[ \frac{R_i}{p_h} \right] \cdot e_h \]
Explicit Accounting for Blocking

\[ R_i = e_i + b_i + \sum_{h<i} \left\lceil \frac{R_i}{p_h} \right\rceil \cdot e_h \]

- Inflation-free blocking analysis
- Response-time analysis with uninflated execution times
- Worst-case blocking bound determined by ILP-solver
- Uninflated execution costs
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Explicit blocking terms ILP formulation
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Explicit blocking terms ILP formulation.
Challenges

Prior analysis is pessimistic due to inflation.

Explicit blocking terms
ILP formulation

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.
So far we only talked about the MSRP...
So far we only talked about the MSRP... 

...but there’s more.
Spin Lock Types

- Non-Preemptable Spinning
- Preemptable Spinning
- FIFO-ordered (MSRP)
- Priority-Ordered
- Unordered
- Priority-Ordered with FIFO tie-breaking
Spin Lock Types

Prior analyses do not generalize to other spin lock types without strong progress guarantees.
Prior Analyses Rely on Strong Progress Guarantees

FIFO-Ordering is analysis-friendly:

Each request can be blocked by at most one request for the same resource from each other processor.
Prior Analyses Rely on Strong Progress Guarantees

FIFO-Ordering is analysis-friendly:

Each request can be blocked by at most one request for the same resource from each other processor.

Prior analyses rely on strong progress guarantees provided by FIFO-ordering.
Each request can be blocked by all other requests for the same resource.
Challenges

Prior analysis is pessimistic due to inflation.

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Explicit blocking terms ILP formulation
Challenges

Prior analysis is pessimistic due to inflation.

Explicit blocking terms
ILP formulation

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Wait-time bounds
Composable constraints
Challenges

Prior analysis is *pessimistic* due to inflation.

Prior analysis is *specific* to non-preemptable FIFO-ordered spin locks.

Explicit blocking terms
ILP formulation

Wait-time bounds
Composable constraints
Analysis of Unordered Spin Locks
Analysis of Unordered Spin Locks

Can all of these really block T1’s request?
Analysis of Unordered Spin Locks

Can all of these really block T1's request?

Observation:
Two requests can only interfere if they were pending at the same time.
Observation:

Two requests can only interfere if they were pending at the same time.
Per-Request Wait-Time Bounds

How many remote request can be pending while T1’s request is pending?
Per-Request Wait-Time Bounds

How many remote request can be pending while T1’s request is pending?

Bound *wait time of* T1’s request. [1]

Bounding the Wait Time of Requests

- Executing
- Spinning
- Critical Section
Bounding the Wait Time of Requests

Executing Critical Section

T2’s response time
Bounding the Wait Time of Requests

worst-case wait time of T1’s request

- Executing
- Spinning
- Critical Section
Bounding the Wait Time of Requests
Bounding the Wait Time of Requests

Executing
Critical Section
Spinning
Bounding
Wait Time
of Requests

\[ X_{2,1,1} \]
\[ X_{2,1,2} \]
\[ X_{2,1,3} \]
\[ X_{3,1,1} \]
\[ X_{3,1,2} \]
Bounding the Wait Time of Requests

Executing Critical Section

X_{2,1,1}, X_{2,1,2}, X_{2,1,3}, X_{3,1,1}, X_{3,1,2}

T1

P1

P2

P3

X_{2,1,1}

X_{2,1,2}

X_{2,1,3}

X_{3,1,1}

X_{3,1,2}

Executing

Spinning

Critical Section
At most one of T2’s jobs can overlap with T1’s request.

Bounding the Wait Time of Requests

- T1
- P1
- P2
- P3
- T2
- T3

- Executing
- Spinning
- Critical Section

EXECUTING
SPINNING
CRITICAL SECTION

t
Bounding the Wait Time of Requests

At most one of T2’s jobs can overlap with T1’s request.

Bound number of conflicting requests.

Executing Critical Section

Spinning

X

X

X

P1

P2

P3

T1

T2

T3

X_{2,1,1}

X_{2,1,2}

X_{2,1,3}
ILP Constraints
Unordered Spin Locks

P1 [T1]

P2 [T2]

P3 [T3]

\[ X_{2,1,1}, X_{2,1,2}, X_{2,1,3} \]

\[ X_{3,1,1}, X_{3,1,2} \]

- Executing
- Spinning
- Critical Section
ILP Constraints
Unordered Spin Locks

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
ILP Constraints
Unordered Spin Locks

\[ X_{2,1,1} + X_{2,1,2} + X_{2,1,3} \leq 1 \]

Same constraints for unordered and FIFO-ordered spin locks.

\[ X_{3,1,1} + X_{3,1,2} \leq 1 \]
Challenges

Prior analysis is pessimistic due to inflation.

Explicit blocking terms ILP formulation

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Wait-time bounds Composable constraints
Prior analysis is pessimistic due to inflation.

Explicit blocking terms ILP formulation

Prior analysis is specific to non-preemptable FIFO-ordered spin locks.

Wait-time bounds Composable constraints
Evaluation
Evaluation

Non-Preemptable Spinning ×

Preemptable Spinning

FIFO-ordered (MSRP)

Priority-Ordered

Unordered

Priority-Ordered with FIFO tie-breaking
Could we reduce pessimism?

Non-Preemptable Spinning

Preemptable Spinning

Priority-Ordered

Unordered

Priority-Ordered (MSRP)

Priority-Ordered with FIFO tie-breaking
Evaluation

Could we **reduce pessimism**?

Does the spin lock type matter at all?

- Preemptable Spinning
- Priority-Ordered (MSRP)
- Priority-Ordered with FIFO tie-breaking
- Unordered
Evaluation

Could we **reduce pessimism**?

Which type should we use?

Does the spin lock type matter at all?

- Preemptable Spinning
- Unordered
- Priority-Ordered with FIFO tie-breaking
Could we **reduce** pessimism?

Does the spin lock type matter at all?

When can **unordered** locks be used?

**Which type** should we use?

FIFO-ordered (MSRP)

Unordered

Priority-Ordered with FIFO tie-breaking
Evaluation

Could we **reduce pessimism**?

Does the spin lock type matter at all?

Which type should we use?

When can **unordered** locks be used?

Should spinning be **preemptable** or **non-preemptable**?
Large-scale schedulability experiments:

- number of processors \( m \): 4, 8, 16
- average task utilization: 0.1, 0.2, 0.3
- critical section lengths: [1\( \mu s \), 15\( \mu s \)], [1\( \mu s \), 100\( \mu s \)]
- number of resources: \( m/2, m, 2m \)
- number of requests: 1, 2, 5, 10, 15
- resource sharing factors: 0.1, 0.25, 0.4, 0.75
Evaluation

Large-scale schedulability experiments:

• number of processors: $m \cdot 4, 8, 16$
• average task utilization: 0.1, 0.2, 0.3
• critical section lengths: $[1\mu s, 15\mu s], [1\mu s, 100\mu s]$
• number of resources: $m/2, m, 2m$
• number of requests: 1, 2, 5, 10, 15
• resource sharing factors: 0.1, 0.25, 0.4, 0.75

1296 different configurations
Evaluation

Large-scale schedulability experiments:

- number of processors: \( m = 4, 8, 16 \)
- average task utilization: 0.1, 0.2, 0.3
- critical section lengths: \([1\mu s, 15\mu s], [1\mu s, 100\mu s]\)
- number of resources: \( r \)
- number of requests: 1, 2, 5, 10, 15
- resource sharing factors: 0.1, 0.25, 0.4, 0.75

1296 different configurations

\( \geq 1000 \) samples per data point in each configuration
Schedulability Experiments

fraction of schedulable task sets

#tasks

less idle time
more contention
Schedulability Experiments

higher is better

fraction of schedulable task sets

#tasks

less idle time
more contention
Notation

\[ F | N \]
Notation

FIFO-ordered

FIN
Notation

F | N

Non-preemptable
## Notation

<table>
<thead>
<tr>
<th>Type</th>
<th>Ordering</th>
</tr>
</thead>
<tbody>
<tr>
<td>FİN</td>
<td>FIFO-ordered non-preemptable</td>
</tr>
<tr>
<td>PİN</td>
<td>Priority-ordered non-preemptable</td>
</tr>
<tr>
<td>UİN</td>
<td>Unordered non-preemptable</td>
</tr>
<tr>
<td>PFİN</td>
<td>Priority-ordered non-preemptable with FIFO tie-breaking</td>
</tr>
</tbody>
</table>
Does the new analysis reduce pessimism?
Does the new analysis reduce pessimism?

![Graph showing schedulable number of tasks against number of tasks for different MSRP classes.]

Configuration: 16 CPUs, avg. utilization: 0.1, 16 shared resources, CS lengths $[1\mu s, 15\mu s]$, at most 2 requests per resource, contention = 0.4
Does the new analysis reduce pessimism?

Configurations: 16 CPUs, avg. utilization: 0.1, 16 shared resources, CS lengths $[1\mu s, 15\mu s]$, at most 2 requests per resource, contention = 0.4
Does the new analysis reduce pessimism?

Config: 16 CPUs, avg. utilization: 0.1, 16 shared resources, CS lengths $[1\mu s, 15\mu s]$, at most 2 requests per resource, contention = 0.4

Does the new analysis reduce pessimism?

Does the new analysis reduce pessimism?

Config: 16 CPUs, avg. utilization: 0.1, 16 shared resources, CS
lengths \([1\mu s, 15\mu s]\), at most 2 requests per resource, contention = 0.4

ILP-based analysis of unordered spin lock

classic analysis

holistic analysis [1]

Does the new analysis reduce pessimism?

Less pessimism with new blocking analysis

Does the Spin Lock Type matter?
Does the Spin Lock Type matter?
Does the Spin Lock Type matter?

Spin lock type has significant impact on schedulability.
When can unordered locks be used?
When can unordered locks be used?
When can unordered locks be used?
When can unordered locks be used?

Task sets schedulable regardless of spin lock type
When can unordered locks be used?

Task sets schedulable regardless of spin lock type

Enables use of simpler algorithms and cheaper hardware!
Handling Unknown Spin Lock Types

Analysis for unordered spin locks makes no ordering assumptions
Handling Unknown Spin Lock Types

Analysis for unordered spin locks makes no ordering assumptions

Analysis for unordered spin locks applicable for unknown types!
What is the impact of allowing preemptable spinning?
What is the impact of allowing preemptable spinning?

Preemptable spinning can...

![Graph showing the impact of preemptable spinning on schedulable tasks.](image)
What is the impact of allowing preemptable spinning?

Preemptable spinning can...

...decrease schedulability (FIFO-ordered)
What is the impact of allowing preemptable spinning?

Preemptable spinning can...

...decrease schedulability (FIFO-ordered)

...increase schedulability (priority-ordered)
Summary and Conclusions
Spin lock type has significant impact on schedulability.

Specify spin lock type
Spin lock type has significant impact on schedulability.

FIFO- and priority ordering required to support many workloads.

Specify spin lock type

Support FIFO and priority ordering in AUTOSAR

Suggested API changes
Spin lock type has significant impact on schedulability.

FIFO- and priority ordering required to support many workloads.

Preemptable spinning can improve schedulability.

Specify spin lock type

Support FIFO and priority ordering in AUTOSAR

Support preemptable spinning with ordering guarantees

Suggested API changes
Novel blocking analysis for spin locks:

- Inflation-free blocking analysis

- Response-time analysis with uninflated execution times
Summary

Novel blocking analysis for spin locks:

\[ \tau \]

inflation-free blocking analysis

\[ \tau \]

response-time analysis with uninflated execution times

support for 8 spin lock types
Novel blocking analysis for spin locks:

- inflation-free blocking analysis

- response-time analysis with uninflated execution times

- support for 8 spin lock types
  - ...including unordered locks: safe for unknown types

Summary
Novel blocking analysis for spin locks:

- inflation-free blocking analysis
- response-time analysis with uninflated execution times

Support for 8 spin lock types:

- ...including unordered locks: safe for unknown types
- asymptotically less pessimistic
Summary

Novel blocking analysis for spin locks:

- Inflation-free blocking analysis

- Response-time analysis with uninflated execution times

- Support for 8 spin lock types

  ...including unordered locks: safe for unknown types

  Asymptotically less pessimistic

- Suggestions for AUTOSAR
Future Work

Current analysis assumes non-nested critical sections.
Future Work

Current analysis assumes non-nested critical sections.

Nested critical sections: work in progress
Fin