Modeling and Verification of Dynamic Command Scheduling for Real-Time Memory Controllers

Yonghui Li¹, Benny Akesson², Kai Lampka³, and Kees Goossens¹

¹Eindhoven University of Technology, the Netherlands,
²CISTER/INESC TEC, ISEP, Portugal,
³Uppsala University, Sweden
yonghui.li@tue.nl
Introduction: Heterogeneous Real-Time System

- Multi-processor systems support hard and soft real-time applications

Diverse memory traffic: size, type, address

Worst-case response time (WCRT)?

Worst-case bandwidth (WCBW)?
Outline

- Background
  - DRAM
  - Dynamically scheduled memory controller
  - Timed automata

- TA modeling of a RT memory controller
  - TA model
  - Property verification

- Experimental results

- Conclusions
DRAM Memories

- DRAM is accessed by scheduling commands
  - ACT, PRE, RD, WR, REF, NOP
  - Subject to timing constraints
  - Bank interleaving
  - Burst count
A transaction is translated into a sequence of commands

- Scheduling algorithm
  - First-Come First-Serve (FCFS) for transactions
  - RD or WR commands have higher priority than ACT
A transaction $T_i (i > 0)$ is executed by scheduling commands to successive banks.
Related Work

- Worst-case analyses of real-time memory controllers are based on analyzing individual timing constraint
  - Applying conservative assumptions -> pessimistic bounds
  - Complex analysis -> time-consuming to derive bounds

- Dataflow modeling of real-time memory controllers [Y. Li, 2015]
  - Dependencies are captured by a dataflow graph
    - Analyzing the dataflow model -> bounds
  - Applying assumptions for unpredictable behavior -> pessimism
  - Providing only worst-case bandwidth bound
Our Proposal

- Model and analysis of real-time memory controllers
  - Modeling with timed automata (TA)
    - Without any assumption
    - Accurate timing analysis
  - TA analysis via model checking with Uppaal

\[
\begin{align*}
\text{Queries} & \quad A[] \text{ not deadlock} \\
& \quad A[] \text{ WCRT} \leq 40
\end{align*}
\]
Timed Automata (TA) Model

- TA essentially model a timed system based on non-deterministic state machines extended with clocks and variables

- An example
  - Timing constraint counter

- In system declaration, the TA template is instantiated to be multiple instances
Outline

- Background
  - DRAM
  - Dynamically scheduled memory controller
  - Timed automata (TA) model
- TA modeling of a RT memory controller
- Experimental results
- Conclusions
Overview of the TA model using different templates

Transaction \[\rightarrow\] Command scheduling: ACT, RD/WR, PRE \[\rightarrow\] Command bus

- Source
  - Trans
  - TDM
    - Bus
      - TDMArb
  - Memory mapping
    - TCC: RW
      - ValidSW
    - TCC: CCD
    - TCC: RCD
- ACT Scheduler
  - ACTBus
  - ValidRAS
  - ValidRP
  - TCC: RAS
  - ValidFAW
  - ACTCmd
- Auto-PRE
  - ValidCCD
  - RWCmd
  - ValidRCD
  - ACTCmd
- RW Scheduler
  - RWCmd
  - RWCmd
  - TCC: FAW
  - ValidRRD
  - TCC: RRD
- Cmd Bus


NRTrans > 0

Technische Universiteit Eindhoven
University of Technology
TA Modeling of a RT Memory Controller

- Intuitive TA Model

Source

Intuitive TA Model

NextTrans?

Trans!

Src

TransType := READ

Trans!

TransType := WRITE

Sink

TDM Bus

Source

Requestor = = 0

TransSize := 256

Requestor = = 1

TransSize := 128

Requestor = = 2

TransSize := 64

Requestor = = 3

TransSize := 32

Requestor = = 4

TransSize := 16

TDM!

End

Memory mapping

Info.BS := BS, Info.BI := BI, Info.BC := BC,
Info.type := TransType, PARQueue[id] := Info,
id := (id+ 1)% MAXNrTrans, NrTrans +

AddrMap!

TDMArb?

Trans?

TDM!

NextTrans?

Trans!

TDM!

Src

TransType := READ

Trans!

TransType := WRITE

TDM!

Requestor := (Requestor+ 1)%MaxReqNr

Id

Bus

Memory

mapping

Requestor := (Requestor+ 1)%MaxReqNr

Requestor := 0

TransSize := 256

Requestor := 1

TransSize := 128

Requestor := 2

TransSize := 64

Requestor := 3

TransSize := 32

Requestor := 4

TransSize := 16

TDMIdle

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!

Trans?

TDM!
TA Modeling of a RT Memory Controller

- Intuitive TA Model

Table 1. Characteristics of the intuitive TA.

| TA instances | 45 |
| clocks       | 40 |
| variables    | 41 |
| locations    | 186 |
| synchronizations | 46 |
| edges        | 226 |
Optimized TA Model

- Multiple timing constraints are captured by a single TA instead of separate TAs.
- We reuse counters for different timing constraints.

#### Table 2. Characteristics of the TA models.

<table>
<thead>
<tr>
<th></th>
<th>Intuitive</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA instances</td>
<td>45</td>
<td>23</td>
</tr>
<tr>
<td>clocks</td>
<td>40</td>
<td>18</td>
</tr>
<tr>
<td>variables</td>
<td>41</td>
<td>55</td>
</tr>
<tr>
<td>locations</td>
<td>186</td>
<td>137</td>
</tr>
<tr>
<td>synchronizations</td>
<td>46</td>
<td>39</td>
</tr>
<tr>
<td>edges</td>
<td>226</td>
<td>186</td>
</tr>
</tbody>
</table>
TA Observers track the time of
- executing a number of transactions -> WCRT Bound
  - Response time is determined by the interfering transactions
- transferring a fixed mount of data -> WCBW Bound
  - The bound on bandwidth replies on the transferred data

\[
\text{A[]} \text{observer.WCRT} \leq \text{Estimate\_Bound}
\]
Experimental Results

- Setup
  - JEDEC-compliant DDR3-1600G SDRAM memory with interface width of 16 bits and a capacity of 2 Gb
  - Uppaal v4.1.19 on a 64-bit CentOS 6.6 system with 24 Intel Xeon(R) CPUs running at 2.10 GHz and with 125 GB RAM
  - Transaction sizes: 16 bytes, 32 bytes, 64 bytes, 128 bytes, and 256 bytes.
Experiment 1: Validation of TA Model

- **Uppaal simulation of the proposed TA model**
  - Input: a sequence of transactions
  - Output: scheduling timings of commands

- **Transaction execution with an open-source cycle-accurate tool RTMemController** [Li et al., ECRTS 2014]
  - [http://www.es.ele.tue.nl/rtmemcontroller/](http://www.es.ele.tue.nl/rtmemcontroller/)

- **Identical scheduling timings of commands are obtained**
  - TA model accurately captures the timing behavior of the memory controller
Experiment 2: Fixed Transaction Size

- **Worst-case response time (WCRT) bound**
  - TA is always equal or better than existing analyses
  - Improvements: max 20% and average 7.7%.
  - Each bound is validated by `RTMemController`
  - $\leq$ 20 minutes, and $\leq$ 7 GB RAM

![Bar chart showing WCRT (cycles) vs. Transaction sizes (bytes) for RD and WR for different transaction sizes: 16 bytes, 32 bytes, 64 bytes, 128 bytes, and 256 bytes. The chart compares analytical, scheduled, and TA methods.](chart.png)
Experiment 2: Fixed Transaction Size

- Worst-case bandwidth (WCBW) bound
  - Improvements: max 25% and average 13.6%
  - $\leq 1.8$ hours and $\leq 15.3$ GB RAM
Experiment 3: Variable Transaction Sizes

- **Worst-case bandwidth (WCBW) bound**
  - With static information, i.e., TDM arbitration, 40% higher WCBW bound
  - \( \leq 6.8 \) hours and \( \leq 30.3 \) GB RAM
Conclusions

- A **timed automata (TA) model** of a real-time memory controller with dynamic command scheduling
  - Public at: [http://www.es.ele.tue.nl/rtmemcontroller/TA.zip](http://www.es.ele.tue.nl/rtmemcontroller/TA.zip)
  - It can be easily **extended** to different memory controllers and SDRAM devices.
- The TA model is **validated** by *RTMemController*
  - The TA model **accurately** captures the timing behavior of the memory controller.
- The TA model achieves better bounds than existing analyses
  - If **static information**, e.g., the TDM arbitration, is given, the verification runs faster and much **better bound** can be obtained.
Thank You.

yonghui.li@tue.nl