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

![Flappy bird](image1) ![H263 decoder](image2) ![motion control](image3)

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

![Diagram of DRAM Memory System]
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
Scheduling Dependencies of a Transaction

- 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

![Diagram of TA model and Model Checker (UPPAAL)]

**Queries**
- $A[]$ not deadlock
- $A[]$ WCRT $\leq 40$

**Results**
- True
- False, diagnostic trace, i.e., counterexample
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
- Command scheduling: ACT, RD/WR, PRE
- Command bus

- Source
  - TDM Bus
  - Memory mapping
  - TDM Arb
  - Address Map
  - Trans
  - NrTrans > 0

- ACT Scheduler
  - ValidRAS
  - ACT Bus
  - ValidRP

- Auto-PRE
  - ValidCCD
  - RWCmd
  - ValidSW

- RW Scheduler
  - ValidFAW
  - RWCmd
  - TCC: RAS
  - ValidRCD

- Memory mapping
  - TCC: RW
  - TCC: CCD
  - TCC: RCD

- Command bus
  - ValidRRD
  - TCC: FAW
  - TCC: RRD
  - TCC: ACTCmd

- Transaction Command bus
TA Modeling of a RT Memory Controller

- Intuitive TA Model

```
Source

NextTrans?
Src
Trans!
Trans!
TransType := READ
TransType := WRITE
Src Sink

TDM Bus

TDMArb?
Requestor = = 0
TransSize := 256
Requestor = = 1
TransSize := 128
Requestor = = 2
TransSize := 64
Requestor = = 3
TransSize := 32
Requestor = = 4
TransSize := 16
Requestor := (Requestor+ 1)%MaxReqNr

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

Memory mapping

AddrMap!
BI := 1, BC := 1
TransSize := 16
BI := 2, BC := 1
TransSize := 32
BI := 4, BC := 1
TransSize := 64
BI := 4, BC := 2
TransSize := 128
BI := 4, BC := 4
TransSize := 256

TDMIdle
TransSize := 16

End

Requestor := (Requestor+ 1)%MaxReqNr
TransSize := 64
TransSize := 32
TDMArb!
Requestor = = 2
Requestor = = 0
Requestor = = 1
Trans?
End
```
TA Modeling of a RT Memory Controller

- Intuitive TA Model

Table 1. Characteristics of the intuitive TA.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA instances</td>
<td>45</td>
</tr>
<tr>
<td>clocks</td>
<td>40</td>
</tr>
<tr>
<td>variables</td>
<td>41</td>
</tr>
<tr>
<td>locations</td>
<td>186</td>
</tr>
<tr>
<td>synchronizations</td>
<td>46</td>
</tr>
<tr>
<td>edges</td>
<td>226</td>
</tr>
</tbody>
</table>
TA Modeling of a RT Memory Controller

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

Table 2. Characteristics of the TA models.

<table>
<thead>
<tr>
<th>TA model</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>
Property Verification of TA Model

- TA Observers track the time of
  - executing a number of transactions \(\rightarrow\) WCRT Bound
    - Response time is determined by the interfering transactions
  - transferring a fixed mount of data \(\rightarrow\) WCBW Bound
    - The bound on bandwidth replies on the transferred data

\[
\begin{align*}
\text{transFinish} &:= \text{false} \\
\text{transFinish} &:= \text{true}, \quad \text{transFinish} := \text{true} \\
\text{transFinish} &:= \text{false} \\
\text{transFinish} &:= \text{false}
\end{align*}
\]

- \(\forall \) observer. WCRT \(\leq\) 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/

- 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*
  - \( \leq 20 \text{ minutes, and } \leq 7 \text{ GB RAM} \)
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
  - ≤ 6.8 hours and ≤ 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