







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 <u>yonghui.li@tue.nl</u>







UPPSALA UNIVERSITET Technische Universiteit **Eindhoven** University of Technology

Where innovation starts

TU

Introduction: Heterogeneous Real-Time System

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



versity of Technology

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



Jniversity of Technology

Dynamically-Scheduled Memory Controller

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





echnische Universiteit

Jniversity of Technology

Related Work

- Worst-case analyses of real-time memory controllers are based on analyzing individual timing constraint

 [H. Kim, 2014, 2015][Y. Krishnapillai, 2014][Y. Li, 2014][J. Reineke, 2011]
 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





Timed Automata (TA) Model

- TA essentially model a timed system based on nondeterministic 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



Intuitive TA Model





ValidFAW[id]!

 $tFAW = = V_FAW$

SAT

Intuitive TA Model





46

226

synchronizations

edges

- Optimized TA Model
 - Multiple timing constraints are captured by a single TA instead of separate TA.

Optimized

23

18

55

137

39

186

Technische Universiteit

University of Technology

Eindhoven

 \succ We reuse counters for different timing constraints.



Property Verification of TA Model

- 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





• *A[] observer.WCRT <= 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]

<u>http://www.es.ele.tue.nl/rtmemcontroller/</u>

- 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
 - \geq 20 minutes, and \leq 7 GB RAM





Experiment 2: Fixed Transaction Size

- Worst-case bandwidth (WCBW) bound
 - Improvements: max 25% and average 13.6%
 - $\blacktriangleright \leq$ 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
 - $\blacktriangleright \leq$ 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: <u>http://www.es.ele.tue.nl/rtmemcontroller/TA.zip</u>
 - 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



