A Requests Bundling DRAM Controller for Mixed-Criticality System

April 23, 2017
RTAS 2017

by: Danlu Guo, Rodolfo Pellizzoni

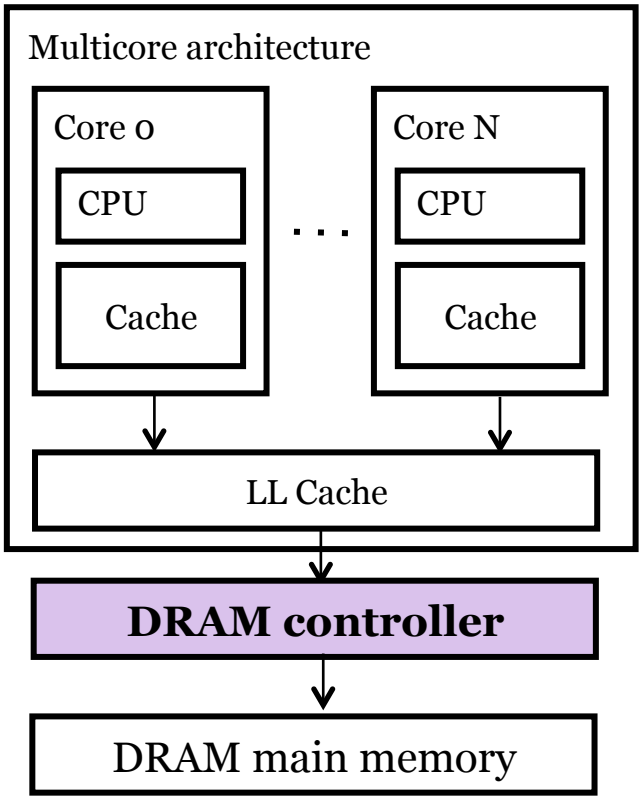
UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

Introduction

- Multicore architecture
  - Shared DRAM main memory
  - Inter-core memory interference
- Real-Time system
  - Hard Real-Time (HRT) applications
  - Soft Real-Time (SRT) applications
- What do we want from DRAM
  - Tighter **upper bound latency** for **HRT** request
  - Better **lower bound bandwidth** for **SRT** request
- Solution:
  - Innovative **predictable** DRAM controllers

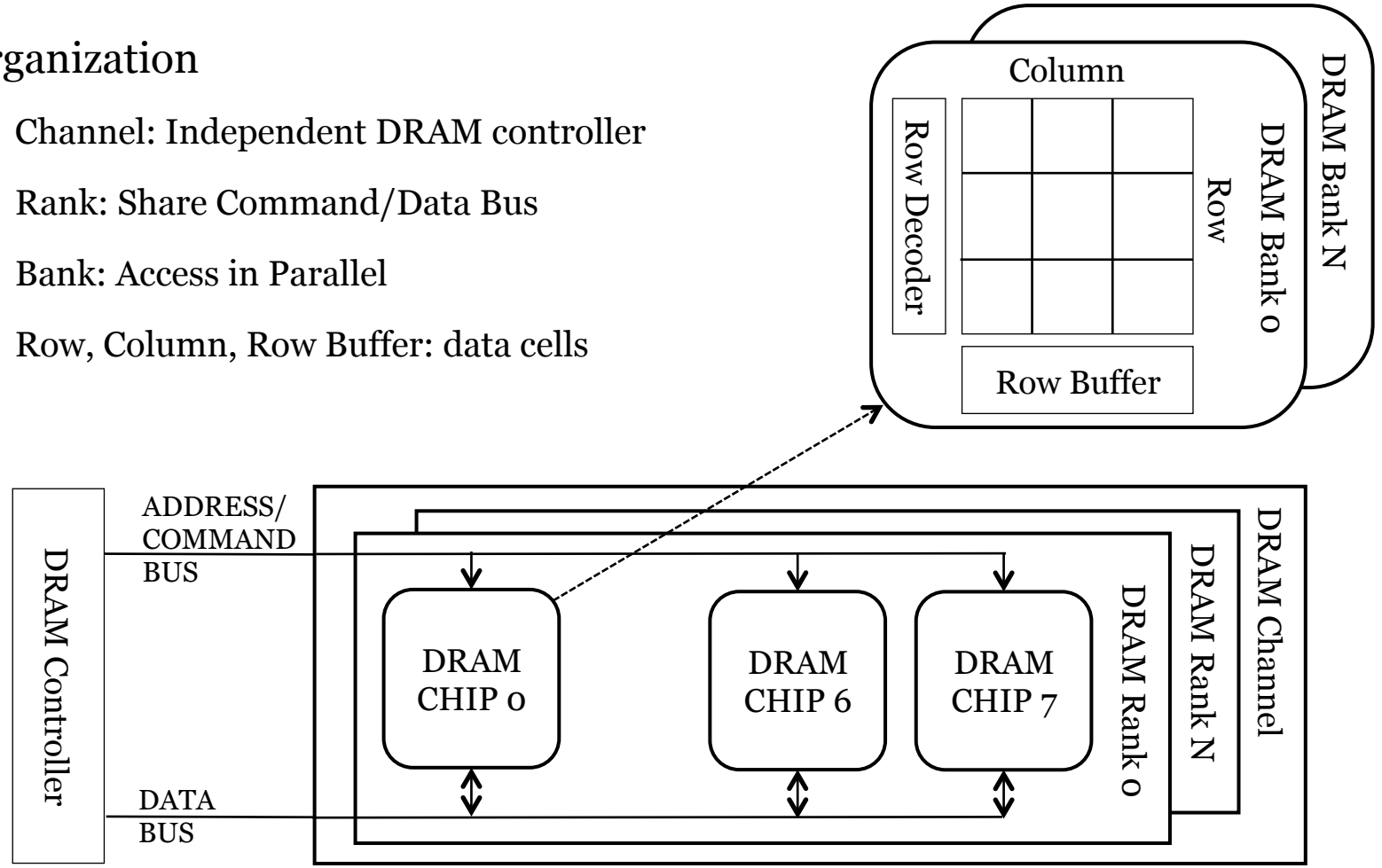


Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

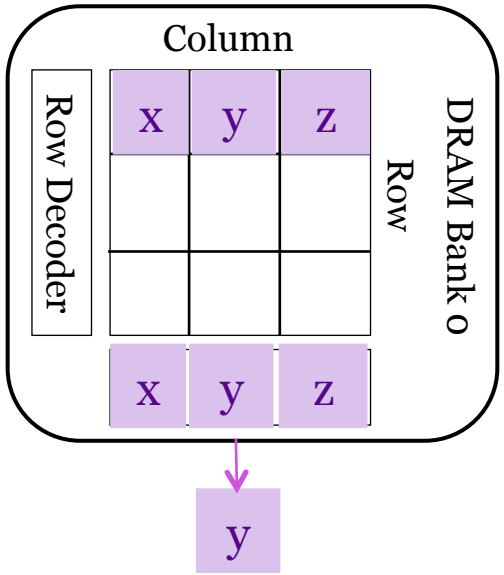
DRAM Background

- Organization
  - Channel: Independent DRAM controller
  - Rank: Share Command/Data Bus
  - Bank: Access in Parallel
  - Row, Column, Row Buffer: data cells

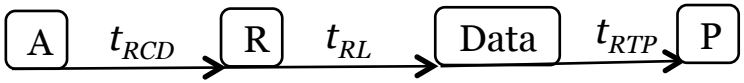


DRAM Background

- Operation
  - Activate (ACT): retrieve data
  - Column-Access-Strobe (RD/WR): access data
  - Precharge (PRE): restore data
  - Timing Constraints (DDR Specifications)



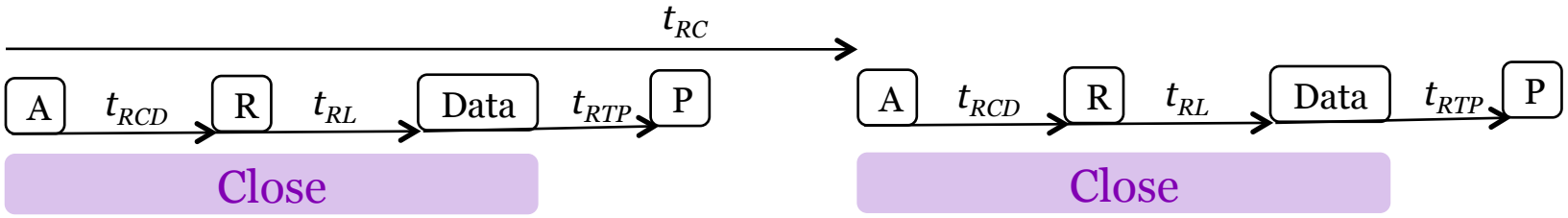
- RD [0,0,1]



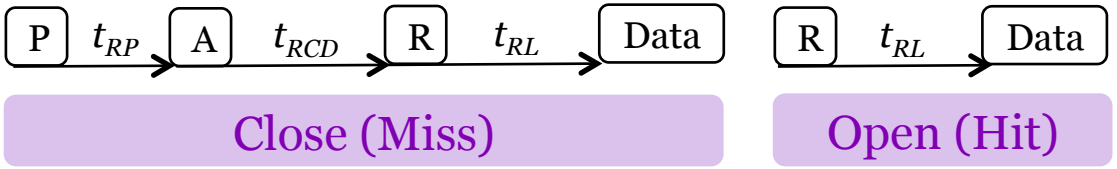
DRAM Background

- Page Policy
  - Close-Page: Precharge (PRE) after access (CAS)

RD[0,0,1], RD[0,0,0]

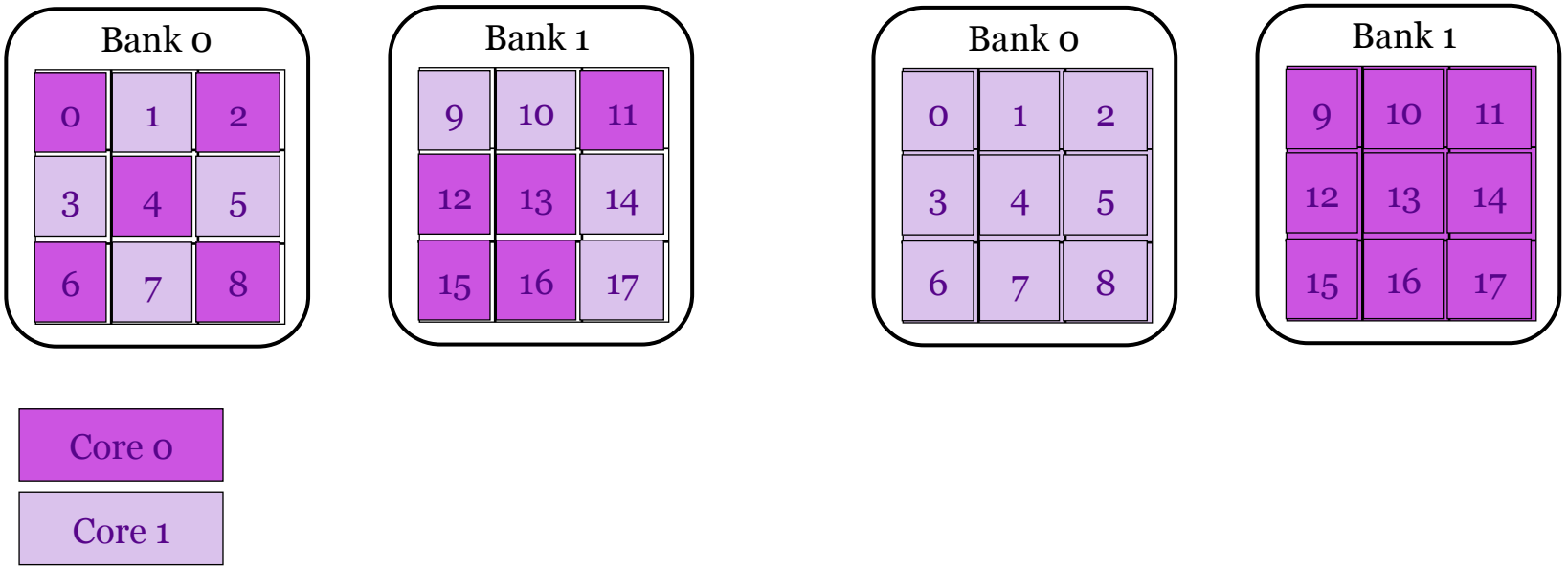


  - Open-Page: Precharge (PRE) when required



DRAM Background

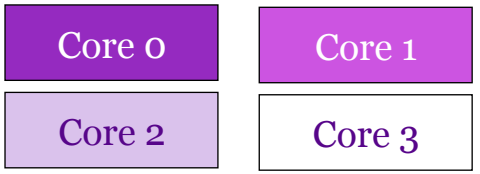
- Data Allocation
  - Shared Banks
    - Allows data sharing among cores
    - Contention on the same bank
  - Private Bank
    - Allows isolation between cores/banks
    - Limits data sharing



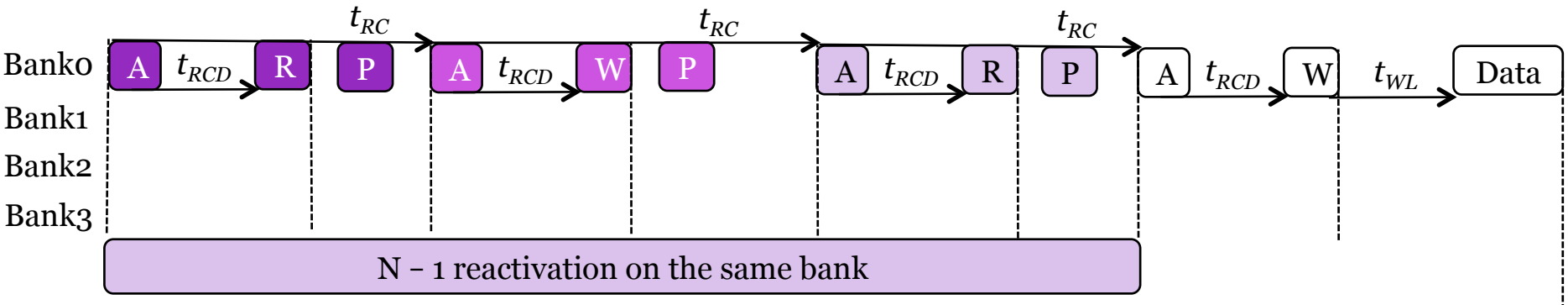
Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

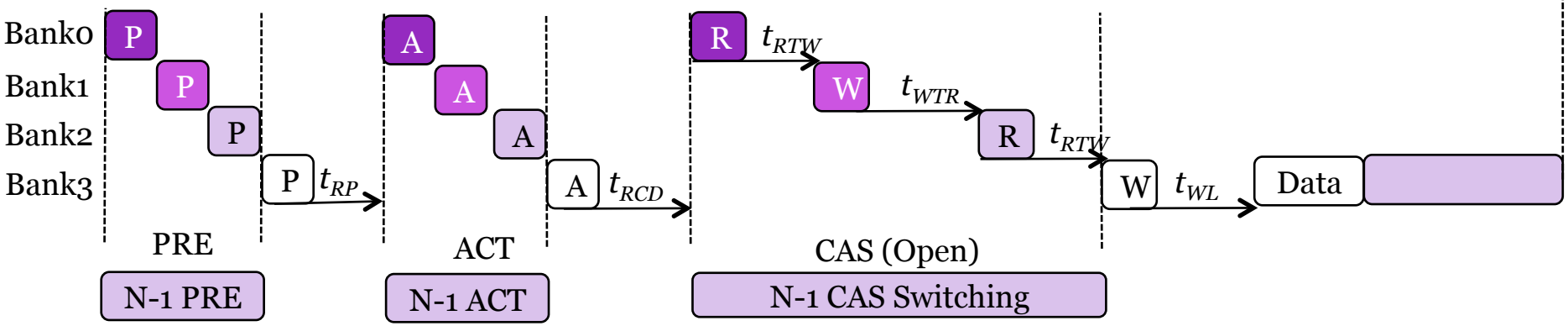
Predictable DRAM Controllers Evaluation



- Shared bank + Close-Page



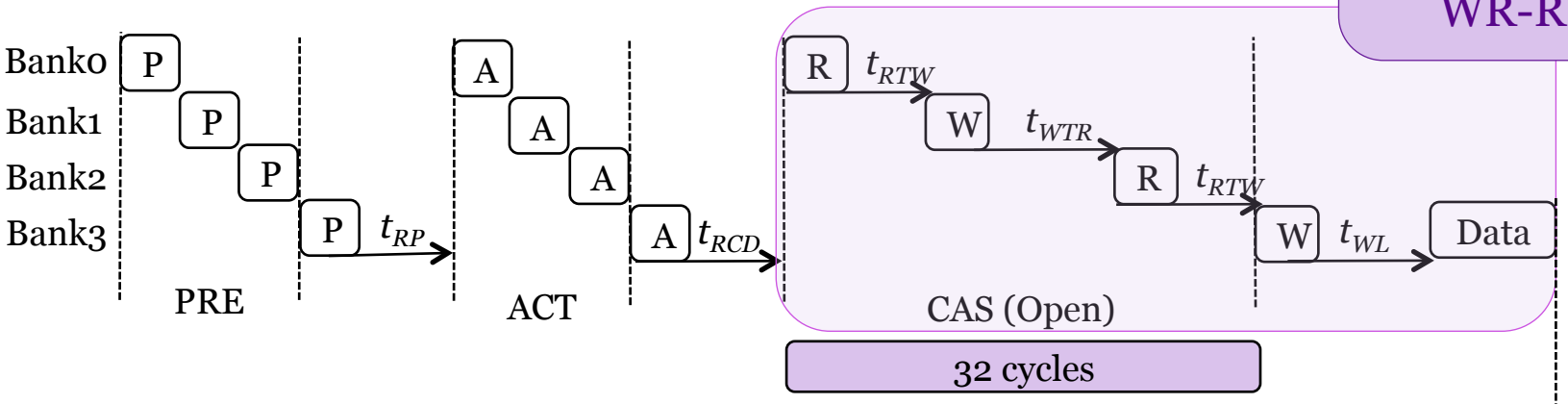
- Private Bank + Open-Page



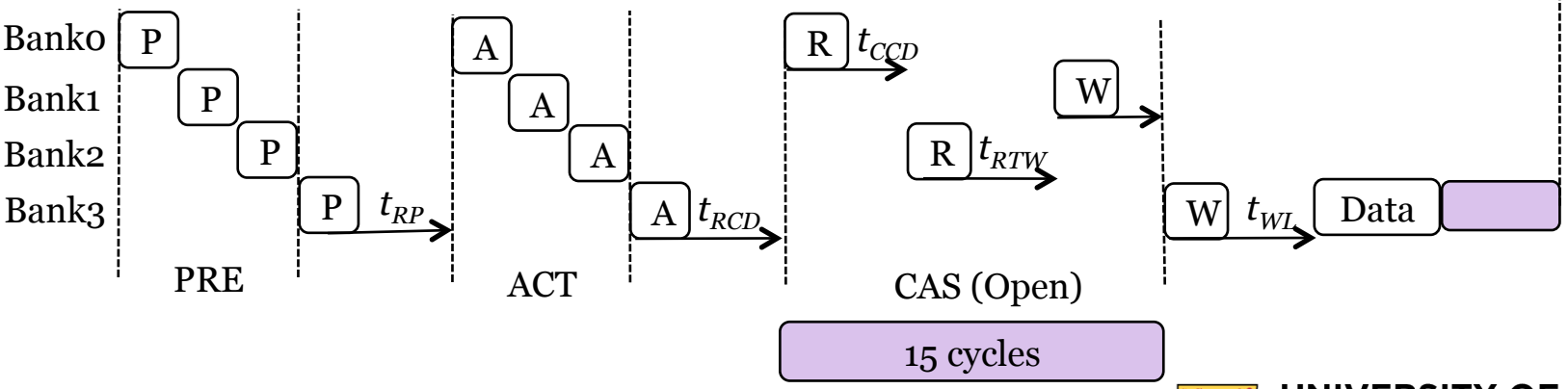
Predictable DRAM Controllers Evaluation

Ex: DDR3-1600H
RD-RD: 4
RD-WR: 7
WR-RD: 18

- Private Bank + Open-Page

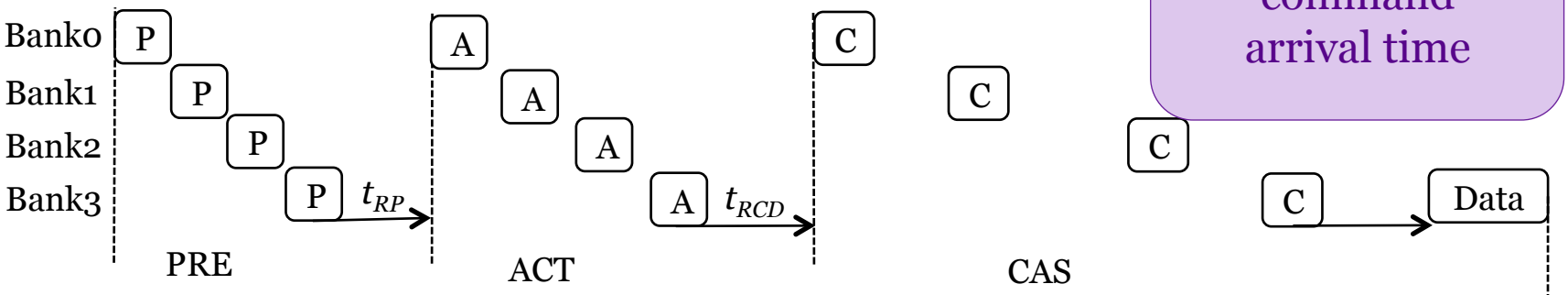


- Private Bank and Open-Page + CAS reordering [L.Ecco & R.Ernst,RTSS'15]



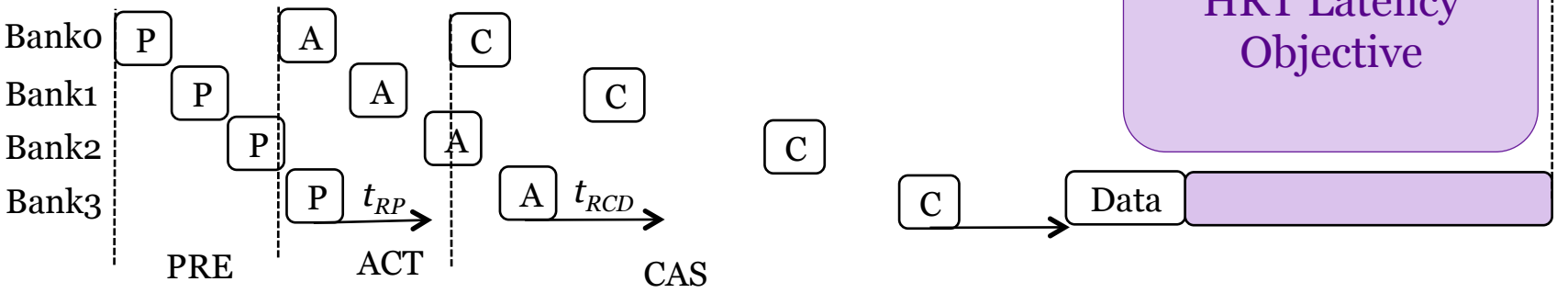
Predictable DRAM Controllers Evaluation

- Current Analytical Model



Not the actual command arrival time

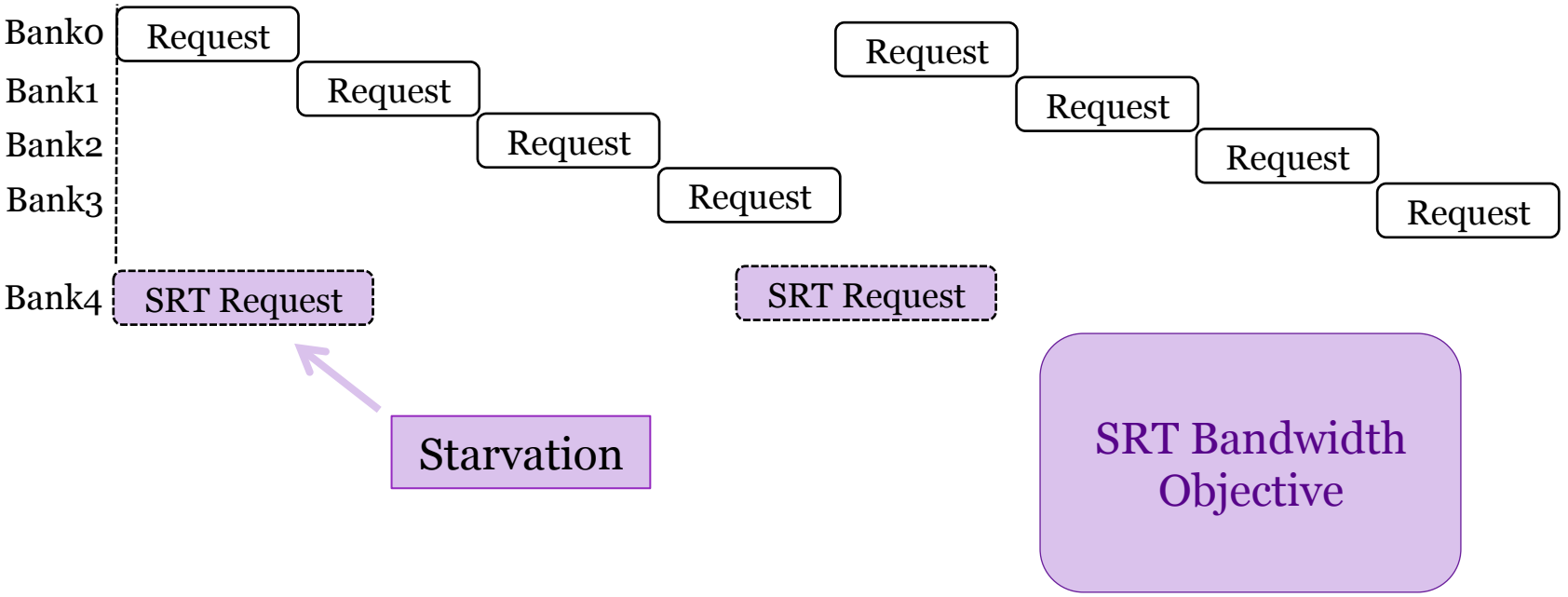
- Pipeline System



HRT Latency Objective

Predictable DRAM Controllers Evaluation

- Mixed Criticality System
  - Co-existing of HRT and SRT applications on different cores
  - Fixed priority can guarantee the HRT latency but limit SRT bandwidth



Objective Summary

- HRT Latency:
  - Apply **Pipelining** can cover the overlap interference.
  - Apply **Reordering** can avoid the repetitive CAS switching.
- SRT Bandwidth:
  - Apply **Co-schedule** of SRT and HRT requests can avoid the starvation.

Reordering CAS breaks the execution sequence

Requests Bundling DRAM Controller

Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Classification
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

Requests Bundling (REQBundle) DRAM Controller

HRT Latency

- Isolation
  - Private bank
- Pipelining and Reordering
  - Close-Page

    => Fixed command sequence
  - Reordering on the request level

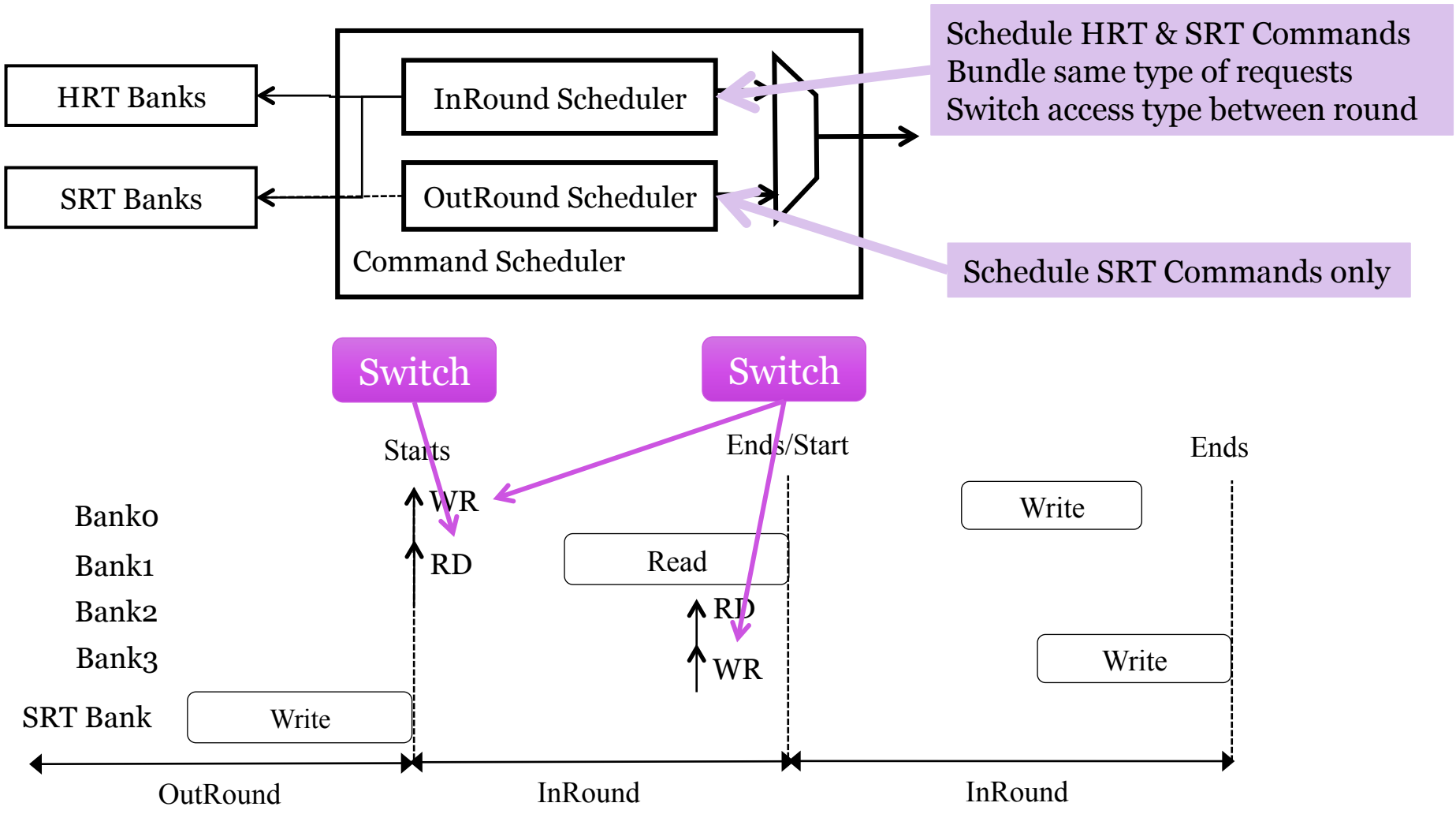
    => Avoid multiple switching

    => Fixed request sequence

SRT Bandwidth

- Fast Access
  - Shared bank + Open-page
- Co-schedule SRT and HRT requests
  - Fixed SRT execution slots before HRT

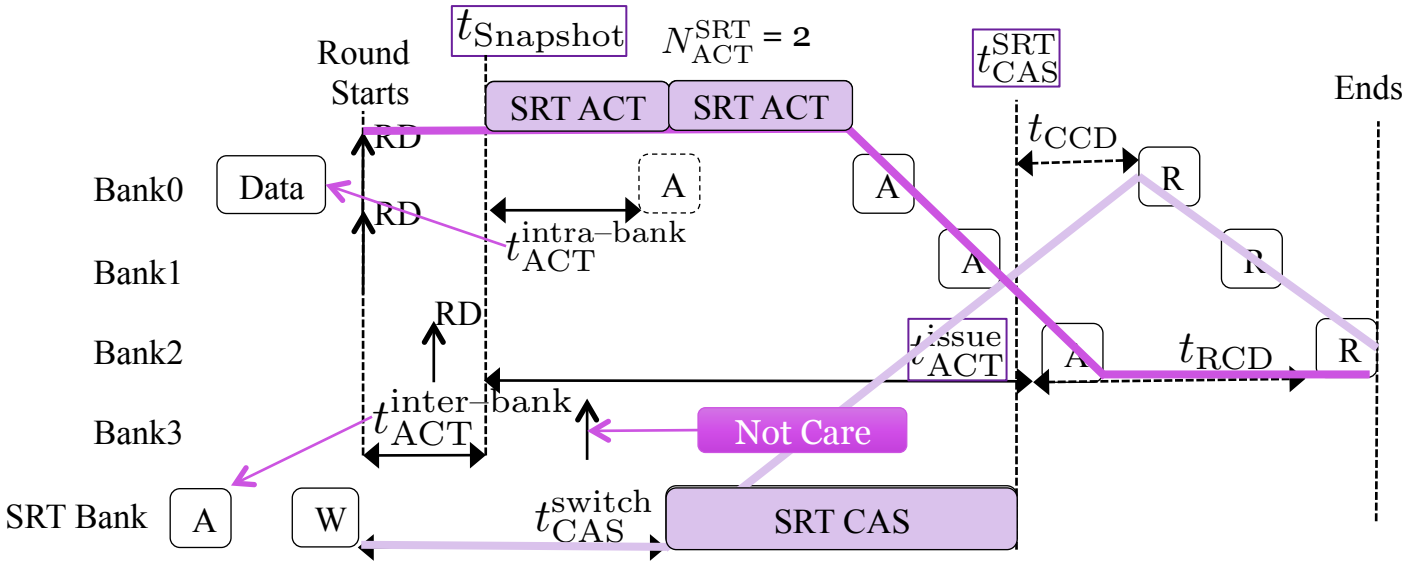
Command Scheduler



UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

InRound Scheduler

- Execution Time of an InRound
  - t_{Snapshot}: time to determine the number of HRT requests (N)
  - $t_{\text{CAS}}^{\text{SRT}}$: time to issue the last SRT CAS
  - $t_{\text{ACT}}^{\text{issue}}$: time to issue the last HRT ACT
  - Execution time R(N) = max($t_{\text{CAS}}^{\text{switch}}$ + (N-1) * t_{CCD} , $t_{\text{ACT}}^{\text{issue}}$ + t_{RCD})

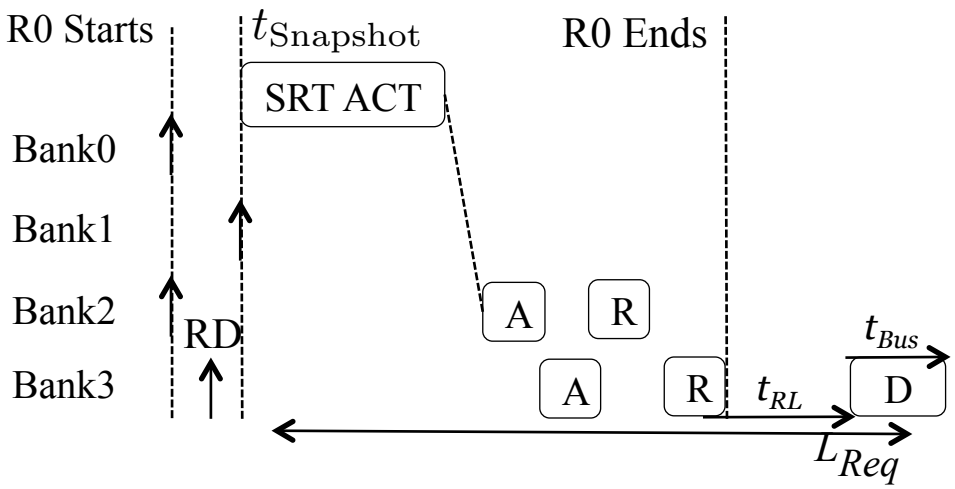


Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

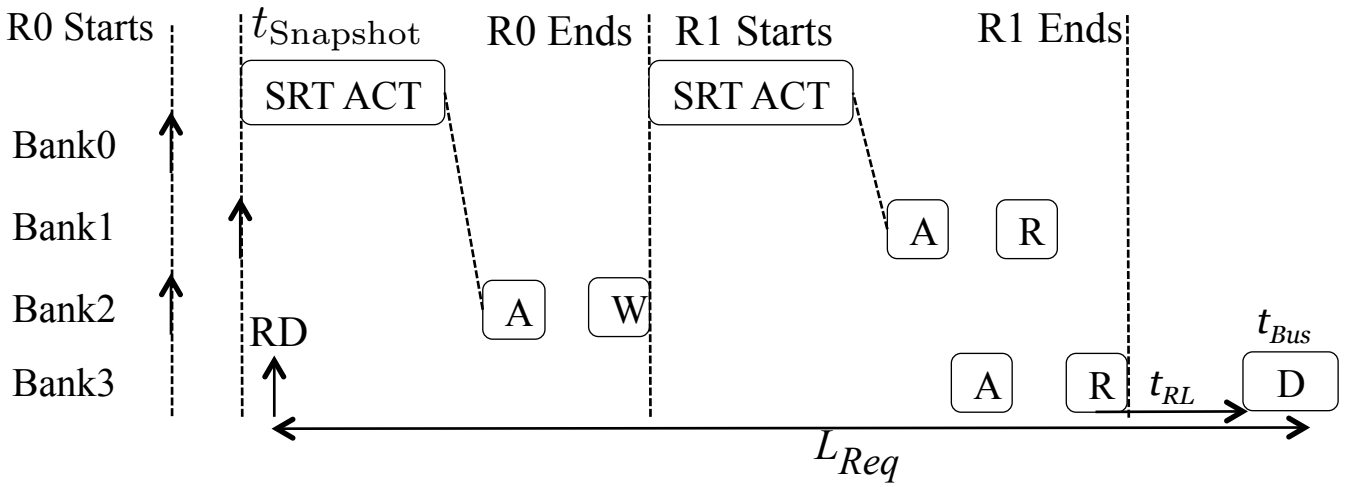
Request Arrival Time and Latency

- Case0: Arrives before snapshot of same type of round
  - L_{Req} = R(No) + tRL + tBus



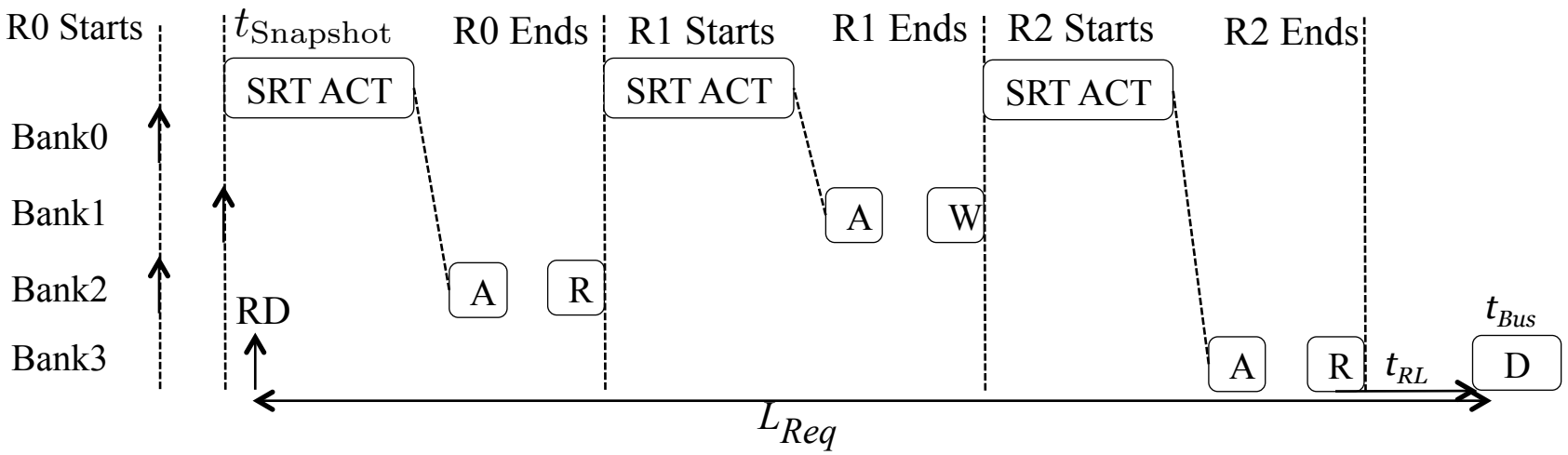
Request Arrival Time and Latency

- Case1: Arrives before/after snapshot of different type of round
  - L_{Req} = R(No) + R(N1) + tRL + tBus



Request Arrival Time and Latency

- Case2: Arrives after snapshot in the same type of round
  - L_{Req} = R(No) + R(N1) + R(N2) + tRL + tBus (Worst Case)



Outline

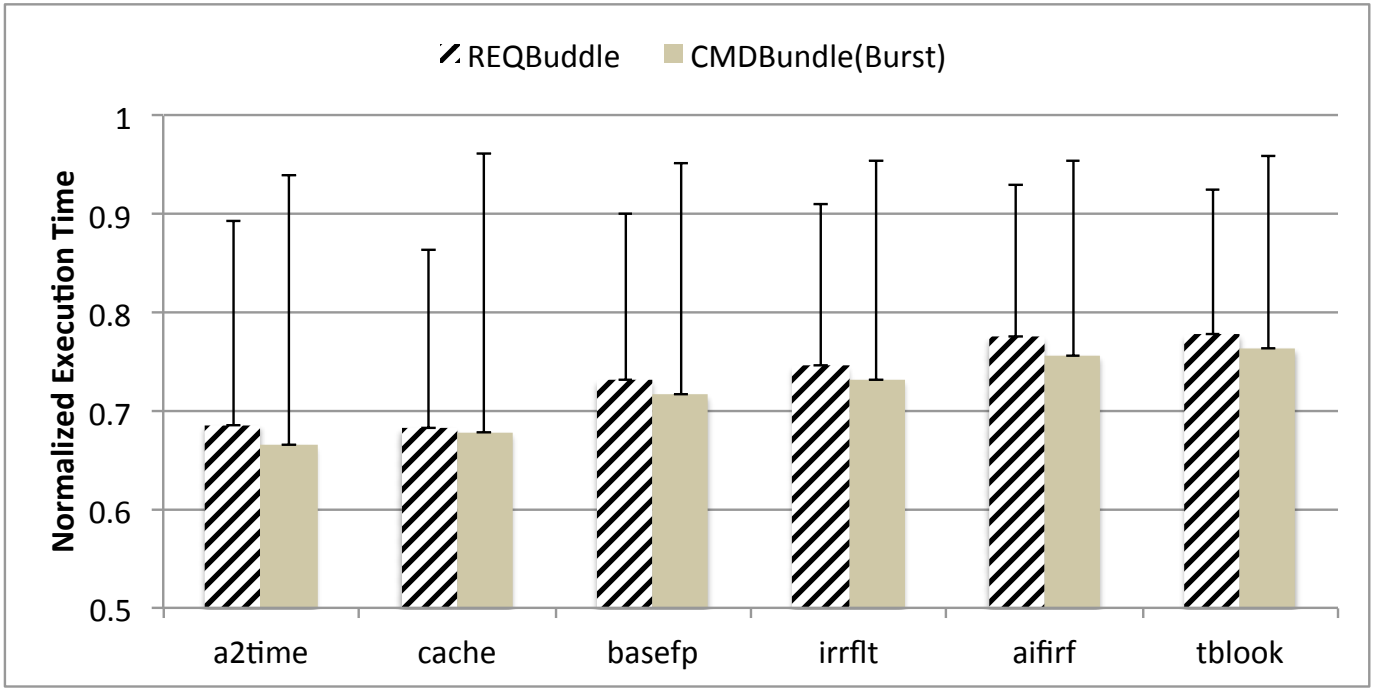
- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

Evaluation

- Implemented in a general DRAM controller simulation framework in C++
  - [DRAMController Demo RTSS'16]
- EEMBC benchmark memory traces generated from MACsim
  - CPU 1GHz
  - Private L1/2 Cache
  - Shared L3 Cache
- Evaluate against Command Bundling (CMDBundle) DRAM Controller
  - [L.Ecco and R.Ernst,RTSS'15]
  - Burst Mode
  - Non-Burst Mode

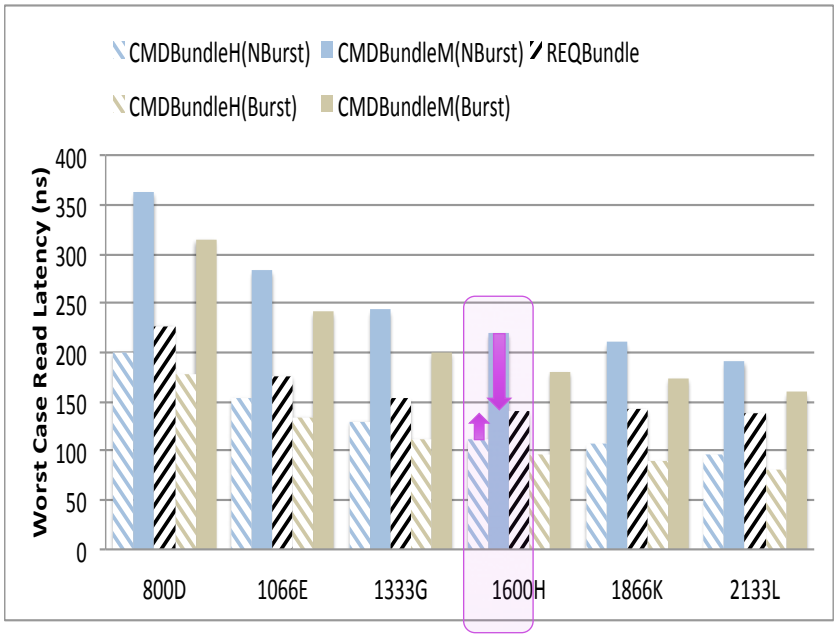
Benchmark Worst Case Execution Time (8 HRTs)

- HRT0 runs benchmark trace and other 7 HRTs run memory intensive traces
- Normalized on CMDBundle (non-burst)

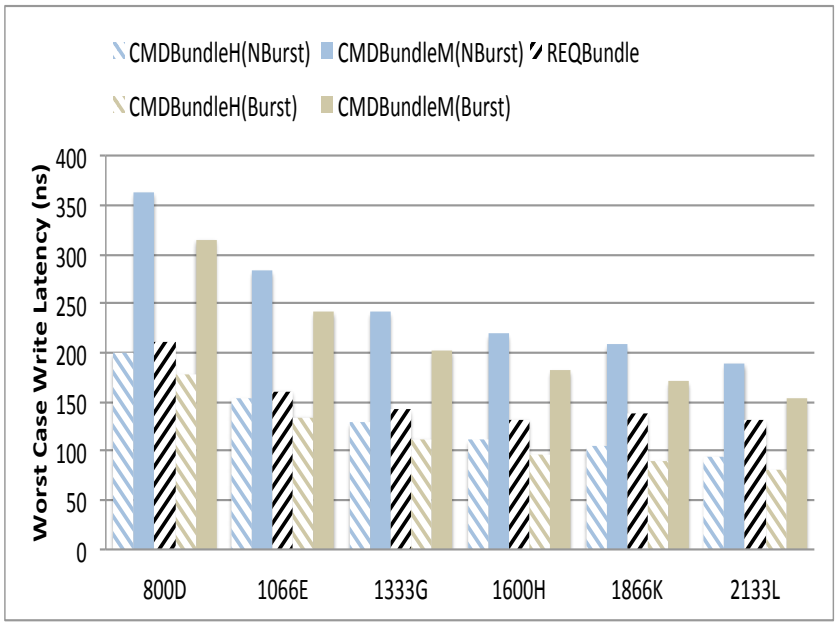


Worst Case HRT Request Latency (8 HRTs)

- RD Request



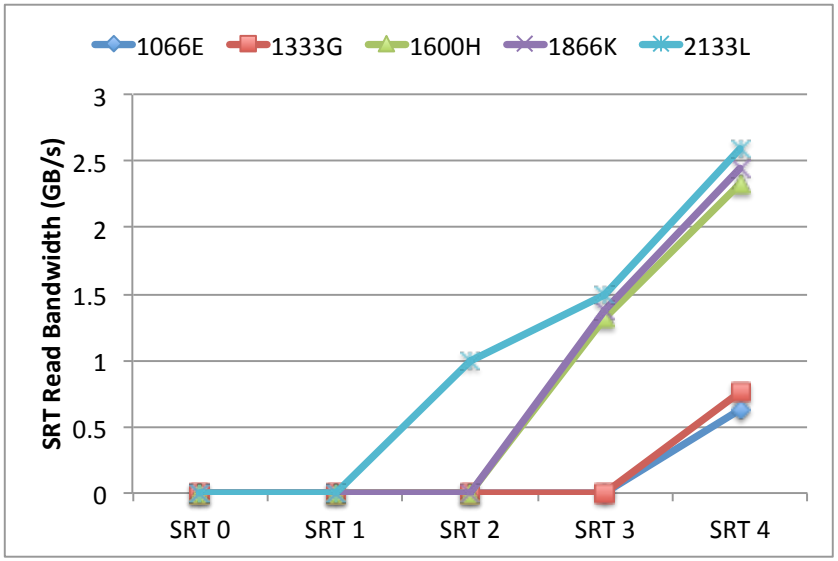
- WR Request



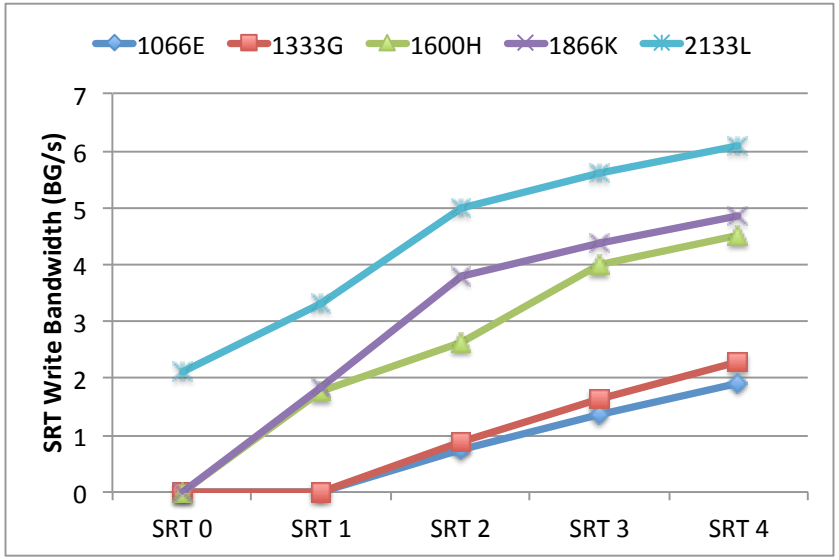
DDR3 Device	800D	1066E	1333G	1600H	1866K	2133L
Read HR(Burst)	0.64	0.61	0.54	0.48	0.37	0.26
Write HR(Burst)	0.76	0.75	0.67	0.58	0.4	0.31
Read HR(NBurst)	0.83	0.83	0.8	0.74	0.65	0.55
Write HR(NBurst)	0.93	0.93	0.88	0.8	0.67	0.61

Worst Case SRT Requests Bandwidth (8 HRTs)

- RD Bandwidth

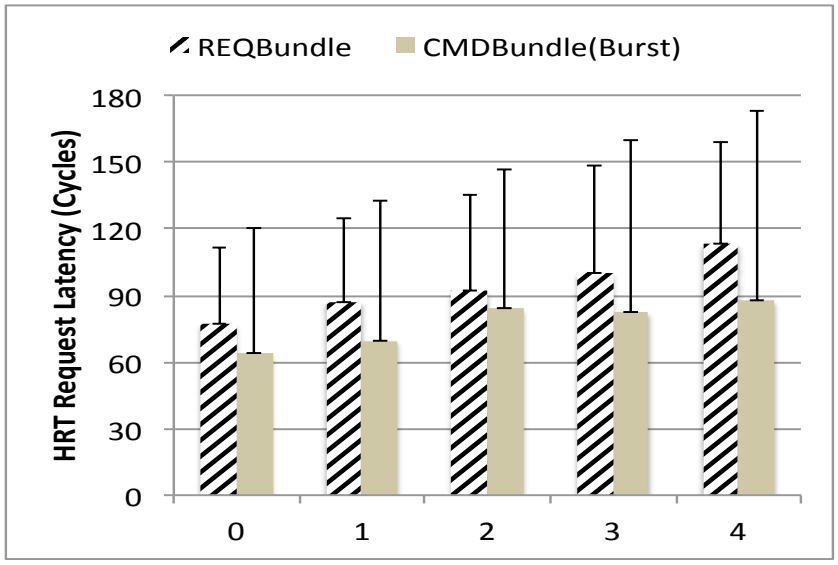


- WR Bandwidth

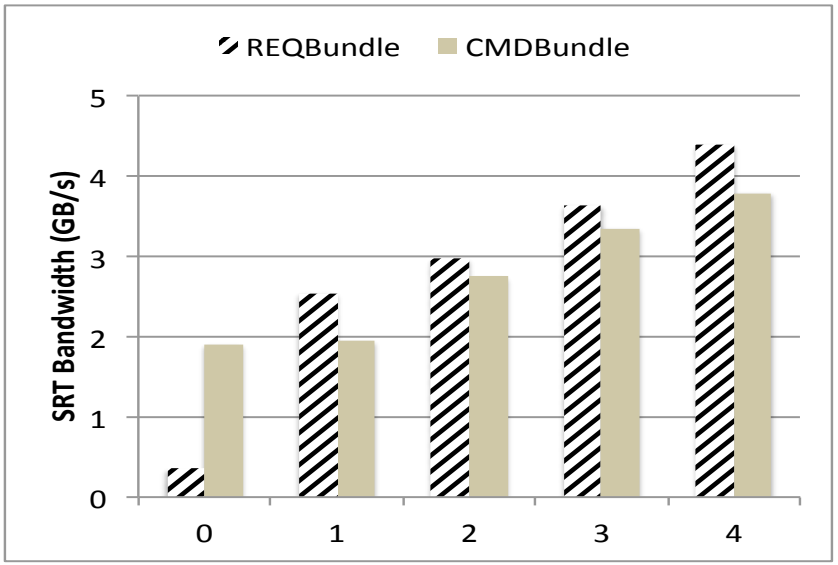


Mixed-Criticality System (8 HRTs, 8 SRTs)

- HRT Latency



- SRT Bandwidth



- Implement virtual HRT requestor mechanism for CMDBundle
  - Considered as a HRT cores in the system
  - All SRT requests share the virtual requestors

Outline

- Introduction
- DRAM Background
- Predictable DRAM Controller Evaluation
- Requests Bundling DRAM Controller
- Worst Case Latency Analysis
- Evaluation
- Conclusion

Conclusion

- Employing request bundling with pipelining can improve the worst case request latency.
- Considering the command timing constraints gaps can provide a good trade-off between the SRT bandwidth and HRT latency.
- Compared with a state-of-the-art real-time memory controller and show the balance point based on the row-hit ratio of a task.
  - Measurement row hit ratio is lower than 50%. A guaranteed row hit ratio requires static analysis and is lower than measured ratio.

UNIVERSITY OF WATERLOO
FACULTY OF ENGINEERING

THANK YOU