Understanding Shared Memory Bank Access Interference in Multi-Core Avionics

Andreas Löfwenmark, Simin Nadjm-Tehrani
Linköping University, Sweden
2016-07-05

LINKÖPING UNIVERSITY

Motivation

- Future safety-critical avionic systems will use multi-core
  - More complex systems => more computational capacity
  - Availability of single-core processors

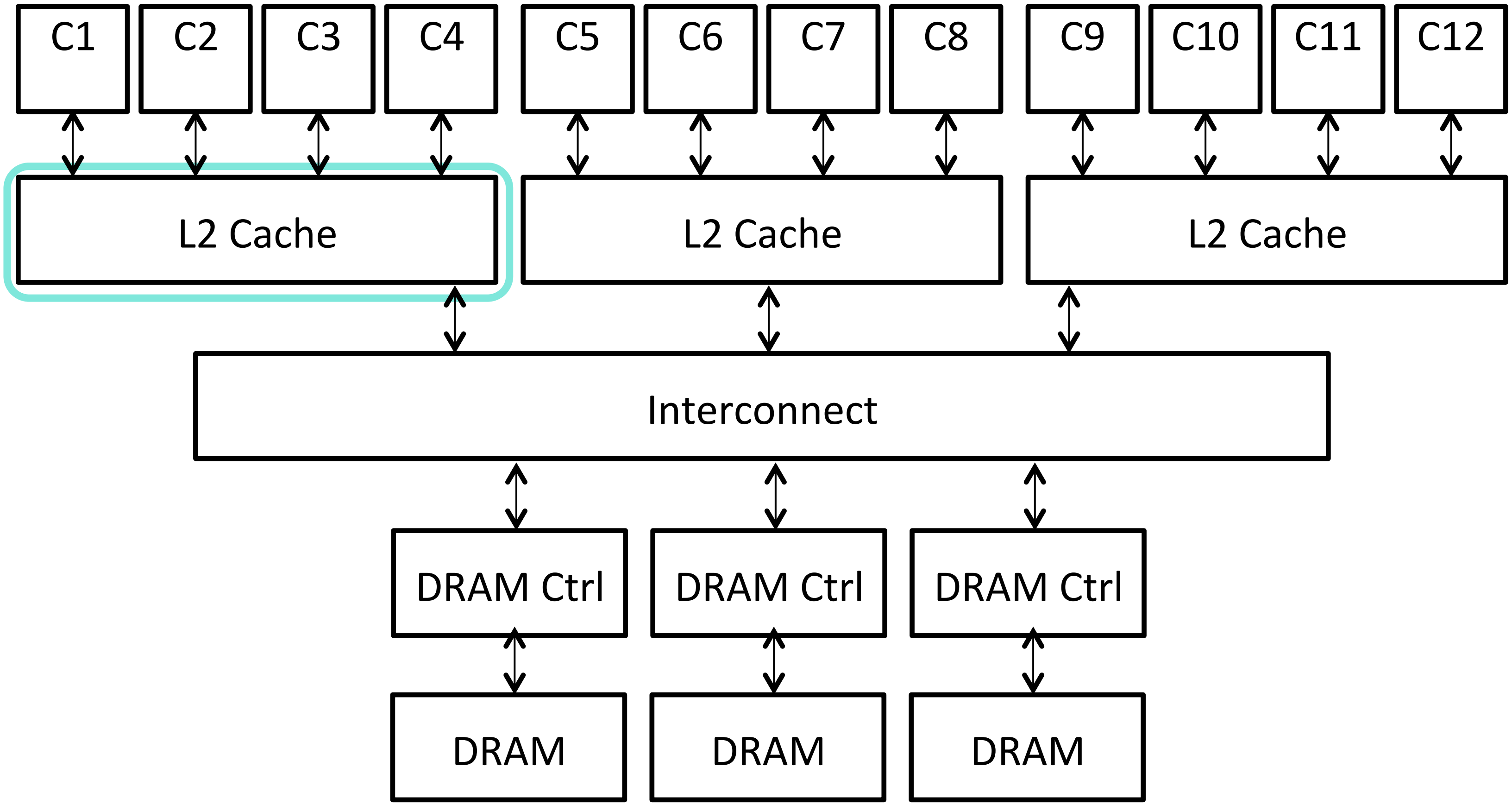
- Demonstrating predictability not yet accomplished

Multi-Core Challenges

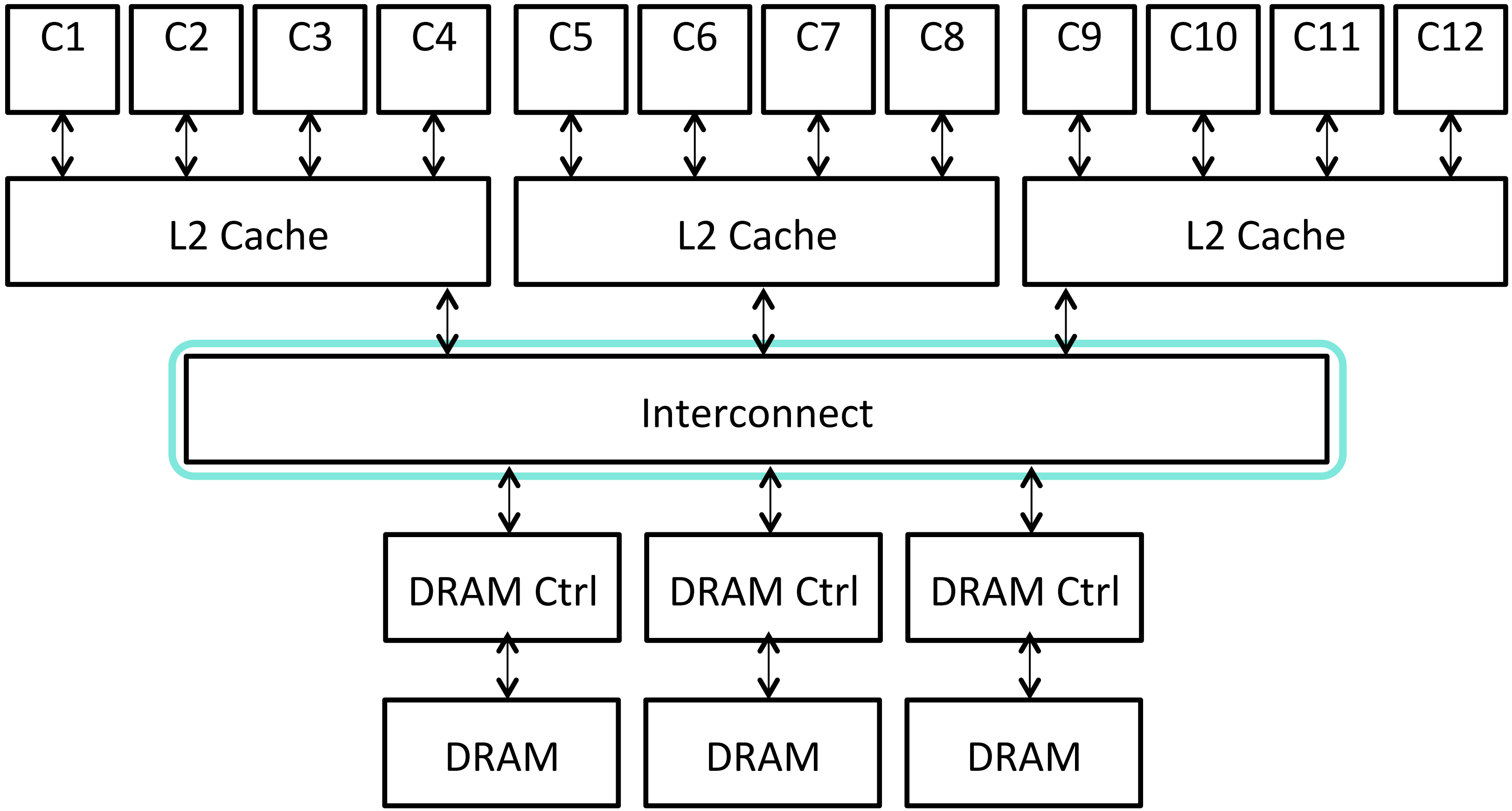
- Achieving Temporal Partitioning
  - Multiple cores can access a shared resource simultaneously

- Worst-Case Execution Time (WCET) and Worst-Case Response Time (WCRT) analysis
  - Pessimism could negate the added processing capacity

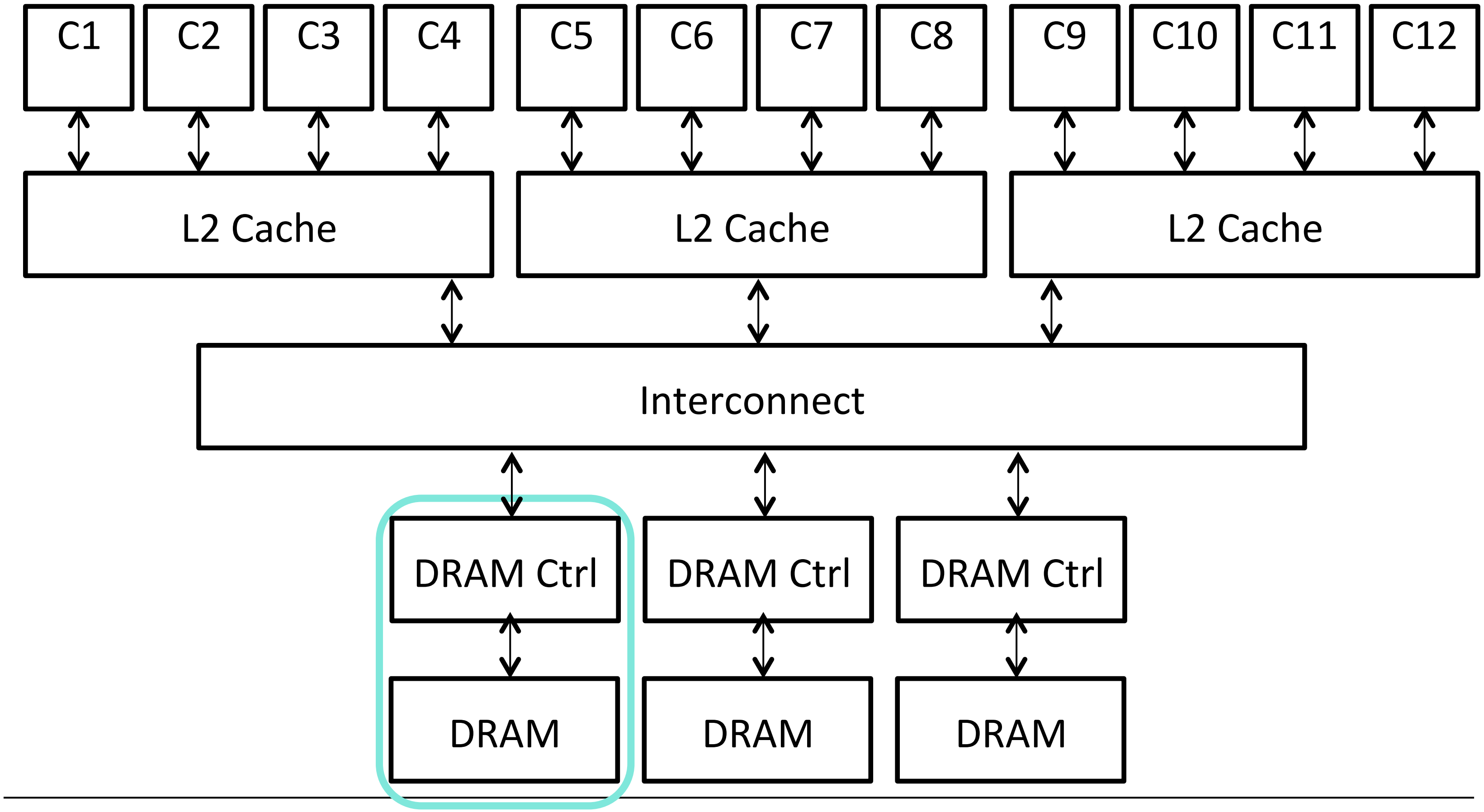
Shared Resources



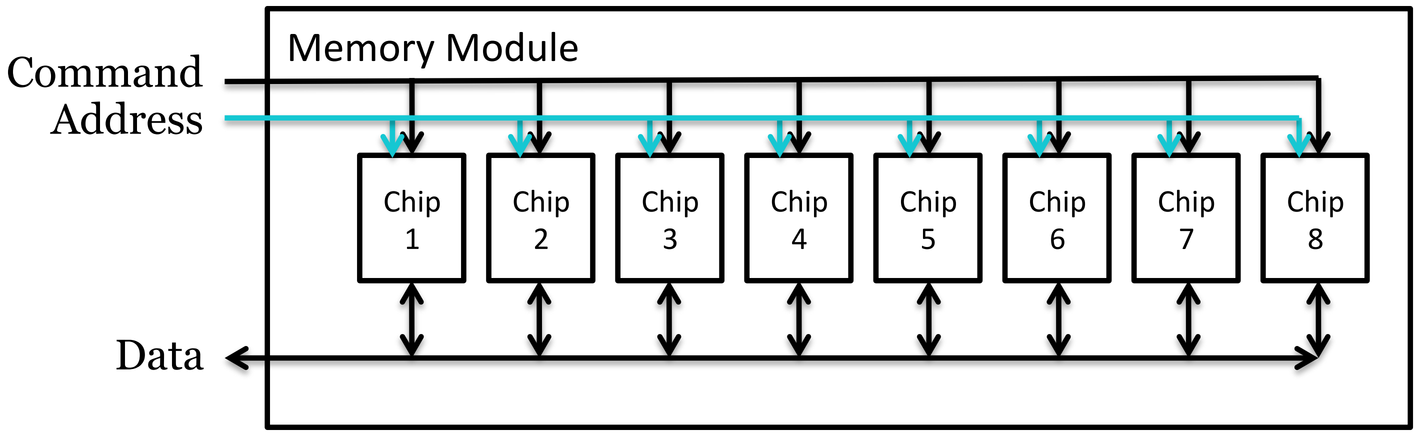
Shared Resources



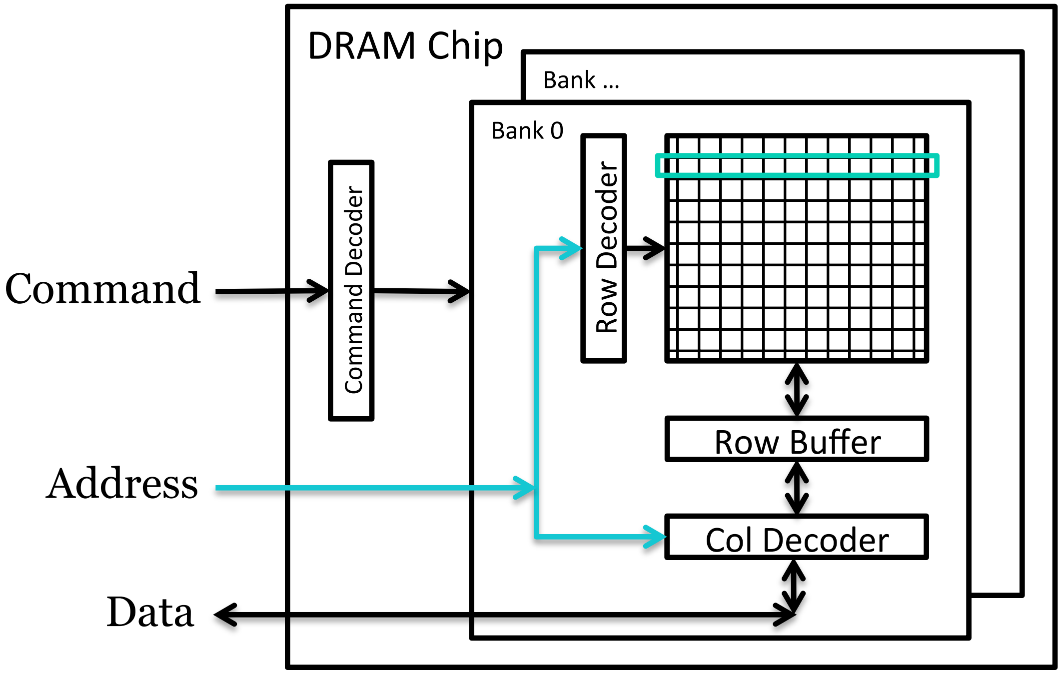
Shared Resources



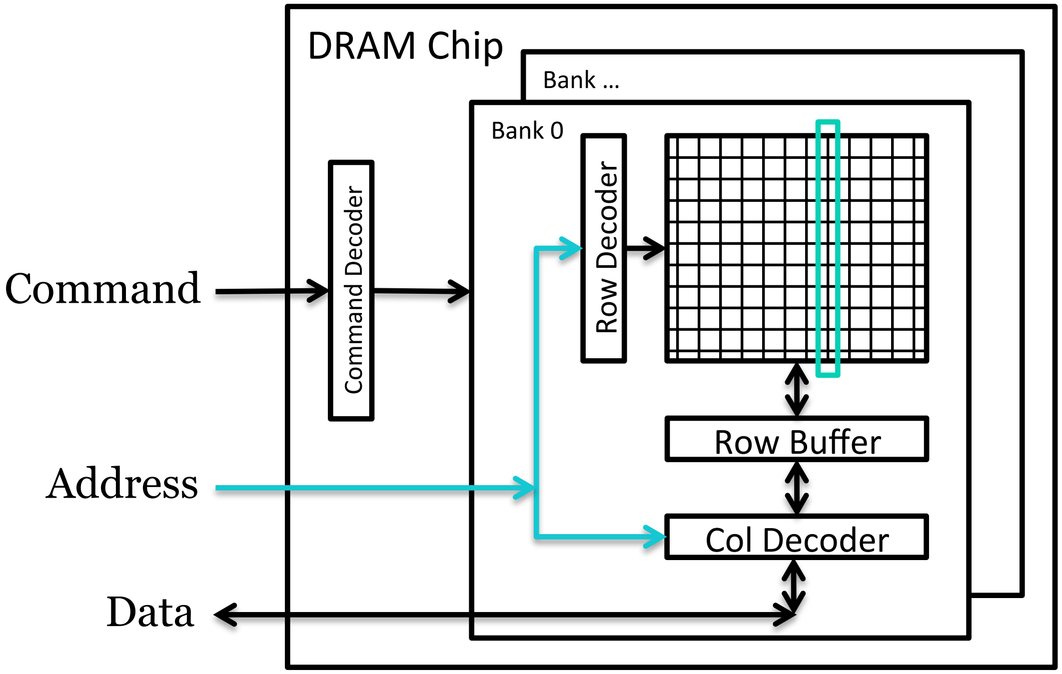
DRAM


Memory Module
Command
Address
Chip 1
Chip 2
Chip 3
Chip 4
Chip 5
Chip 6
Chip 7
Chip 8
Data

DRAM

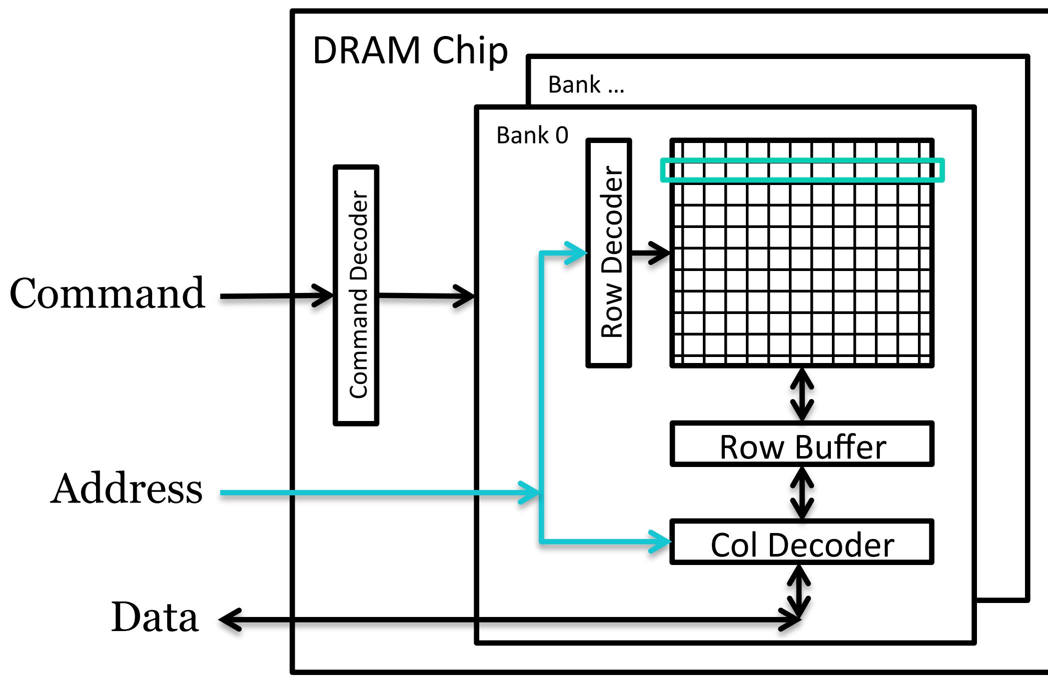

DRAM Chip
Bank ...
Bank 0
Command Decoder
Row Decoder
Row Buffer
Col Decoder
Command
Address
Data

DRAM


DRAM Chip
Bank ...
Bank 0
Command
Command Decoder
Row Decoder
Row Buffer
Address
Col Decoder
Data

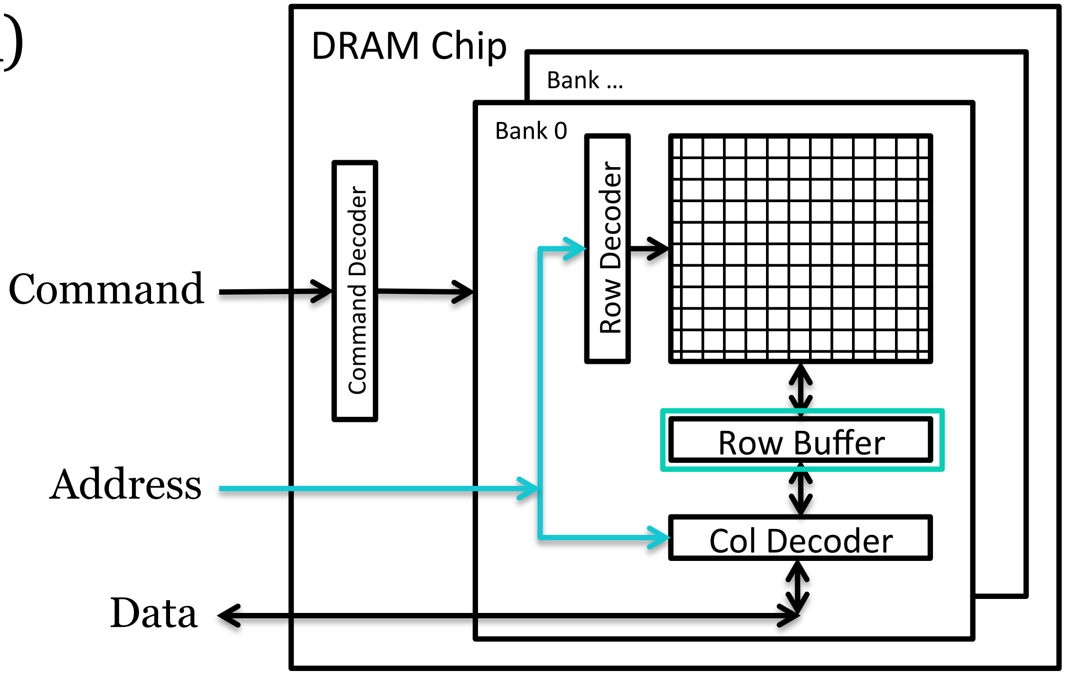
DRAM

- Activate (ACT)



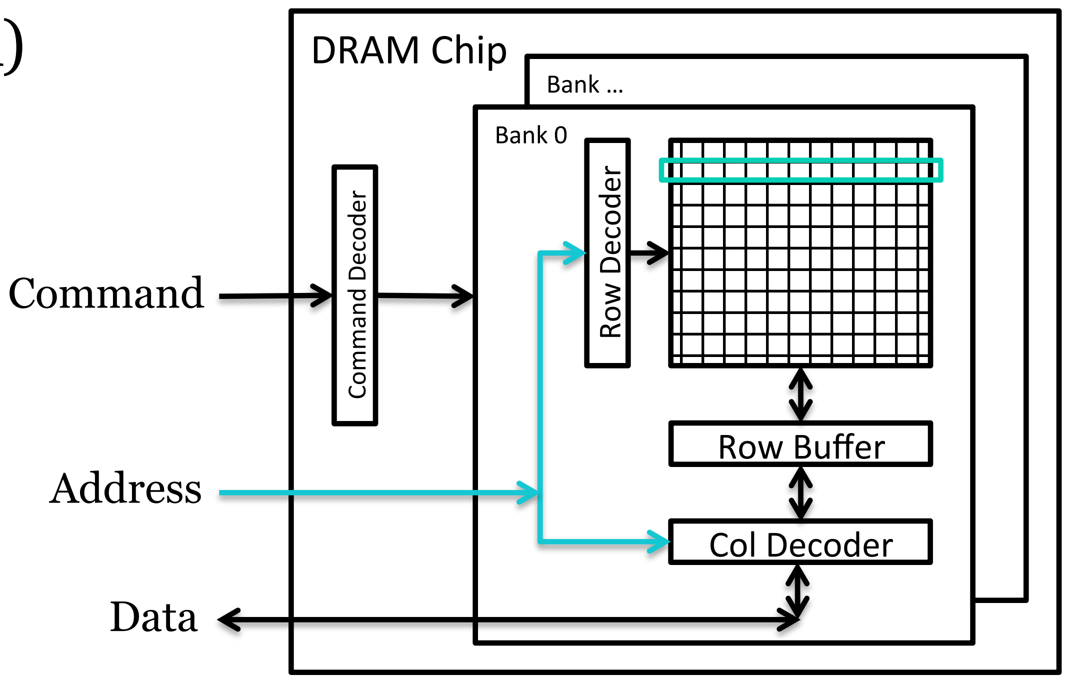
DRAM

- Activate (ACT)
- Read/Write (RD/WR)



DRAM

- Activate (ACT)
- Read/Write (RD/WR)
- Precharge (PRE)



Contributions

- Single Core Equivalence (SCE) based WCRT estimation (Mancuso et al., 2015) combined with shared bank interference delay (Kim et al., 2014)
- Validation of WCRT estimates
  - Understanding (shared) bank interference
  - Comparison of estimates with measurements
- Adaptation of avionics RTOS

SCE (Mancuso et al., ECRTS 2015)

- Colored Lockdown
  - Colors memory pages and locks them in cache
- MemGuard
  - Limits the number of memory accesses per core
- PALLOC
  - Allocates memory in specified DRAM banks for each core

Adapting WCRT Analysis

- Why?
  - Private banks not always feasible
  - Not viable on our RTOS

- Change estimation equations to allow shared banks

WCRT Estimation (Kim et al., RTAS 2014)

- Classical response time test is extended

$$R_i^{k+1} = C_i + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot C_j + H_i \cdot RD_p + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot H_j \cdot RD_p$$

H_i : Number of memory requests for task i

RD_p : Interference delay per request from core p

WCRT Estimation (Kim et al., RTAS 2014)

- Classical response time test is extended

$$R_i^{k+1} = C_i + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot C_j \boxed{+ H_i \cdot RD_p} + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot H_j \cdot RD_p$$

H_i : Number of memory requests for task i

RD_p : Interference delay per request from core p

WCRT Estimation (Kim et al., RTAS 2014)

- Classical response time test is extended

$$R_i^{k+1} = C_i + \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot C_j + H_i \cdot RD_p \boxed{+ \sum_{\tau_j \in hp(\tau_i)} \left\lceil \frac{R_i^k}{T_j} \right\rceil \cdot H_j \cdot RD_p}$$

H_i : Number of memory requests for task i

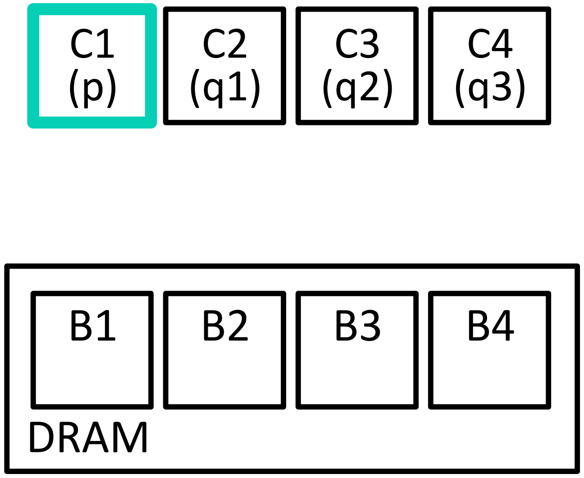
RD_p : Interference delay per request from core p

Contributions

- Single Core Equivalence (SCE) based WCRT estimation (Mancuso et al., 2015) combined with shared bank interference delay (Kim et al., 2014)
- Validation of WCRT estimates
  - Understanding (shared) bank interference
  - Comparison of estimates with measurements
- Adaptation of avionics RTOS

Interference Delay

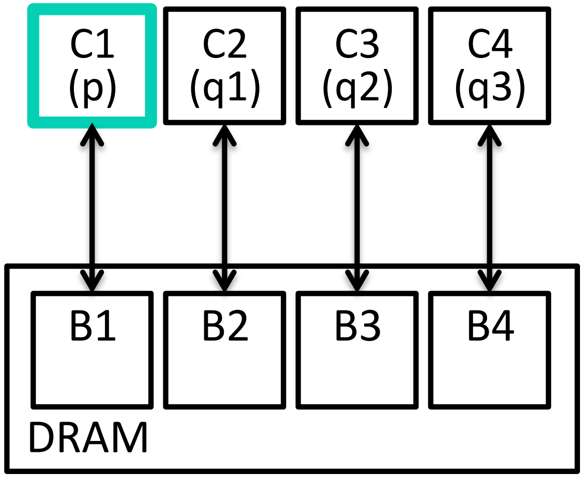
$$RD_p = RD_p^{\text{inter}} + RD_p^{\text{intra}}$$



Interference Delay

$$RD_p = RD_p^{\text{inter}} + RD_p^{\text{intra}}$$

$$RD_p^{\text{inter}} = \sum_{\substack{q \neq p \\ \text{shares no} \\ \text{bank with p}}} (L^{PRE} + L^{ACT} + L^{RW})$$

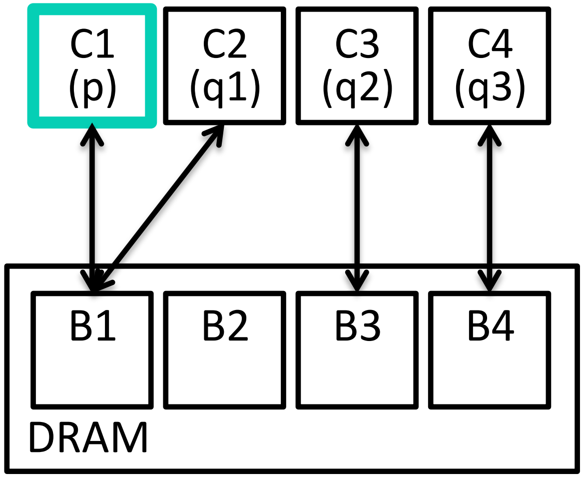


Interference Delay

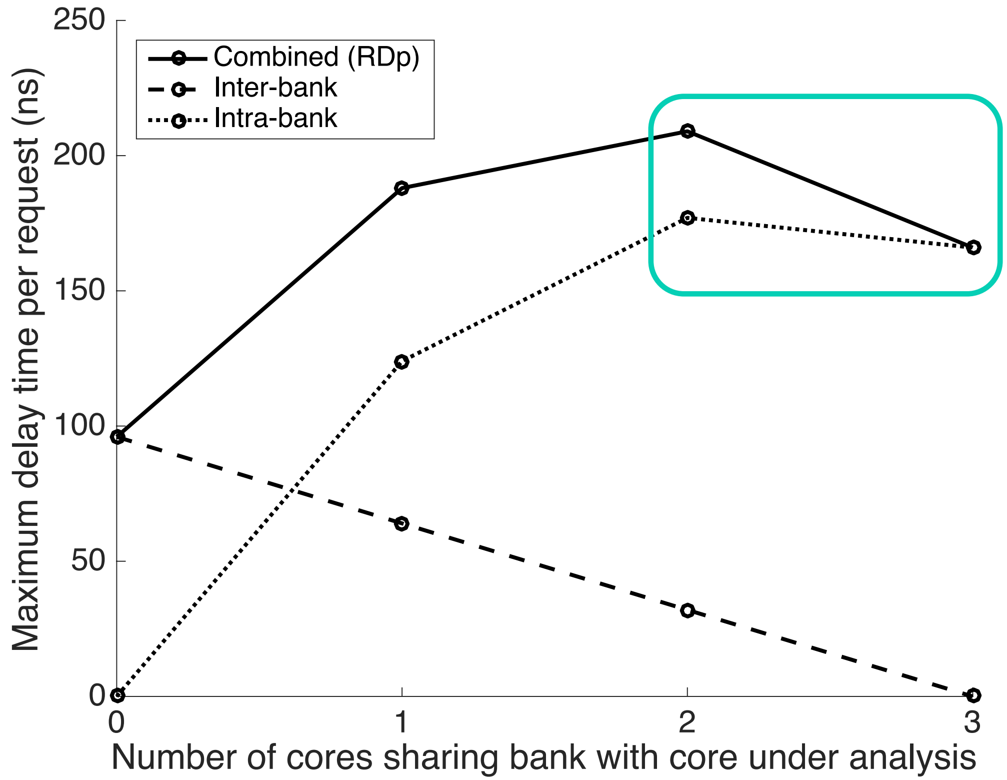
$$RD_p = RD_p^{\text{inter}} + RD_p^{\text{intra}}$$

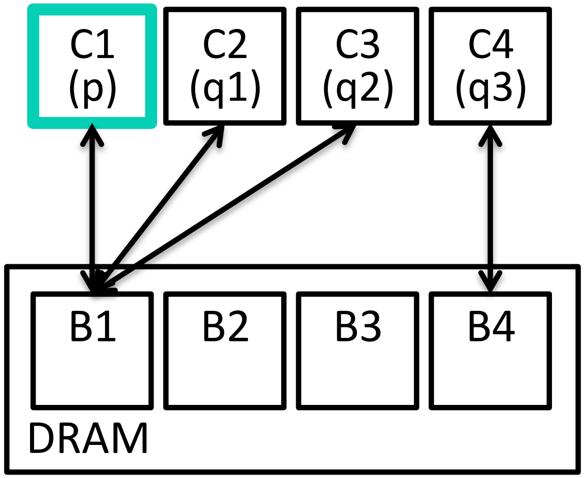
$$RD_p^{\text{inter}} = \sum_{\substack{q \neq p \\ \text{shares no} \\ \text{bank with p}}} (L^{PRE} + L^{ACT} + L^{RW})$$

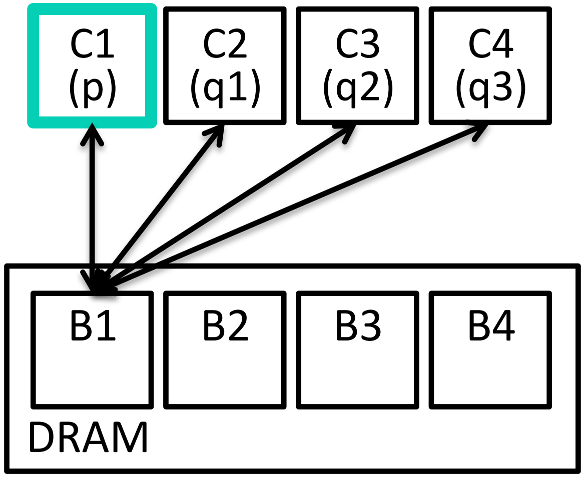
$$RD_p^{\text{intra}} = reorder(p) + \sum_{\substack{q \neq p \\ \text{shares bank} \\ \text{with p}}} (L_{conf} + RD_q^{\text{inter}})$$



Interference Delay







Contributions

- Single Core Equivalence (SCE) based WCRT estimation (Mancuso et al., 2015) combined with shared bank interference delay (Kim et al., 2014)
- Validation of WCRT estimates
  - Understanding (shared) bank interference
  - Comparison of estimates with measurements
- Adaptation of avionics RTOS

Validation of the Adapted Model

- Calculate WCRT
  - Estimate single-core WCET in isolation (C)
    - Measurement based
  - Estimate number of memory requests (H)

- Perform multi-core experiments to measure WCRT

Estimations on Single-Core

- Estimate single-core WCET in isolation (C)
- Count number of memory requests (H)

Partition	WCET (C) (us)	Memory Requests (H) (Partition)	(RTOS)
Nav	14	93	54
Mult	16615	21740	160
Cubic	9345	45	38
Image	4391	560	40

Comparison of Calculated and Measured

- Calculate WCRT
- Measure WCRT

Partition	Core	Period (us)	Response Time (R) (us) Estimated	Measured
Nav	0	16667	45	14
Mult	1	16667	21192	16620
Cubic	2	16667	9362	9345
Image	3	16667	4516	4391

Focusing on processes with tight margins

- Methodology
  - Measure WCRT with a memory intensive synthetic application on core 1 – 3
  - With and without memory regulation

Partition	Core	Period (us)	Response Time (R) (us) No regulation	Regulation
Mult	0	16667	17075	16654

  - "Right" memory access restriction here can also be used as bound when run with other applications!

Conclusions

- Adaptation of SCE framework and avionics RTOS indeed support running critical avionic applications on multi-core

- Bound on interference delay with shared DRAM banks is a pessimistic upper bound – as expected

- Insight: Worst-case interference need not arise in a scenario with maximum number of cores

Andreas Löfwenmark, Simin Nadjm-Tehrani
Linköping University, Sweden

www.liu.se

LINKÖPING UNIVERSITY