Memory Controller Performance for Smartphone Workloads

Goran Narančić

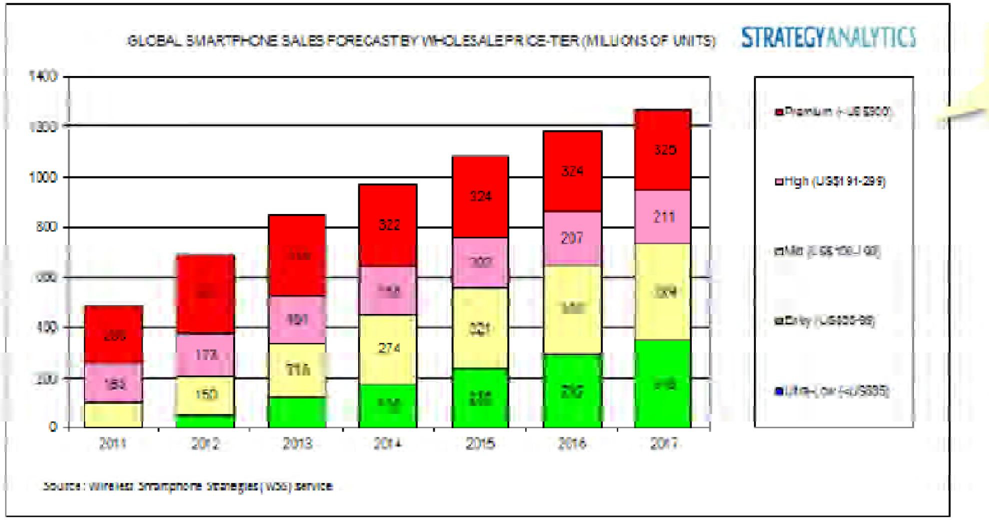
IEEE Croatia Section RL 07 Technical Talk

Zagreb, February 14. 2013.

Motivation: Smartphone Market

The Largest Market Ever

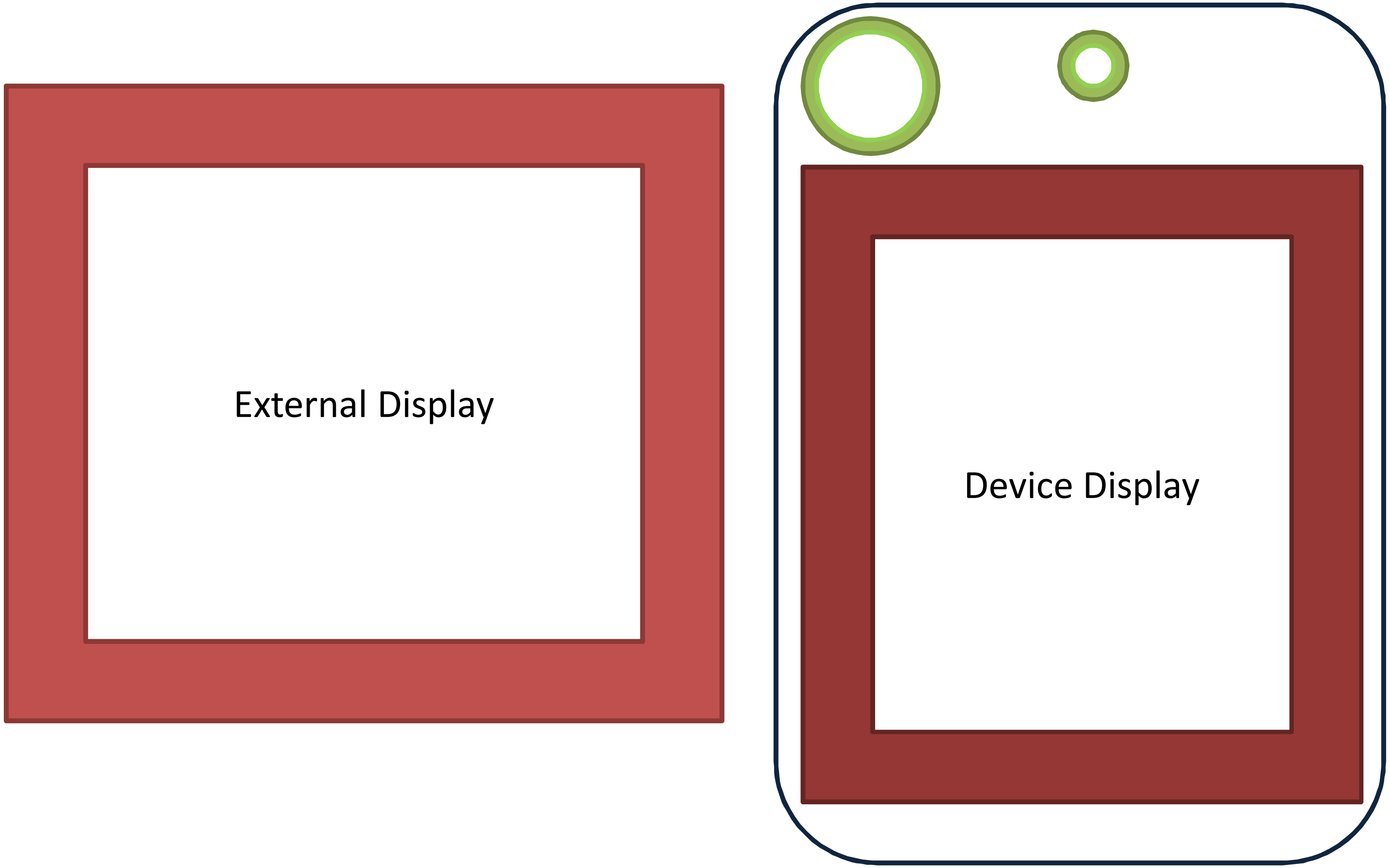
- IDC - 1.8 billion mobile phones will ship in 2012
  - By the end of 2016, 2.3 billion mobile phones will ship per year



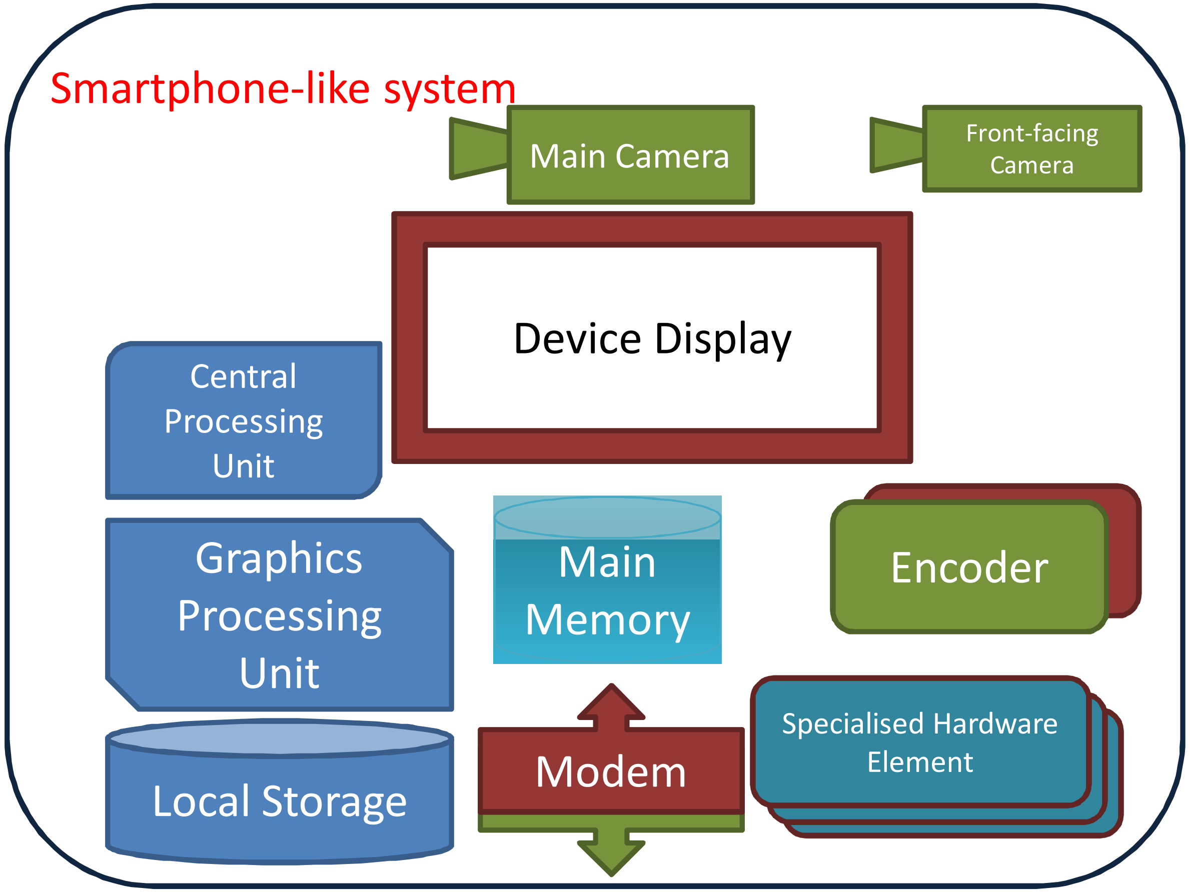
Source: Wireless Smartphone Strategies (WSS) service

Smart phones accounting for approximately half of total phone market

Smartphone Systems



Smartphone Systems



- strict and competing requirements: energy vs. real-time performance
- are conventional solutions appropriate?

This work

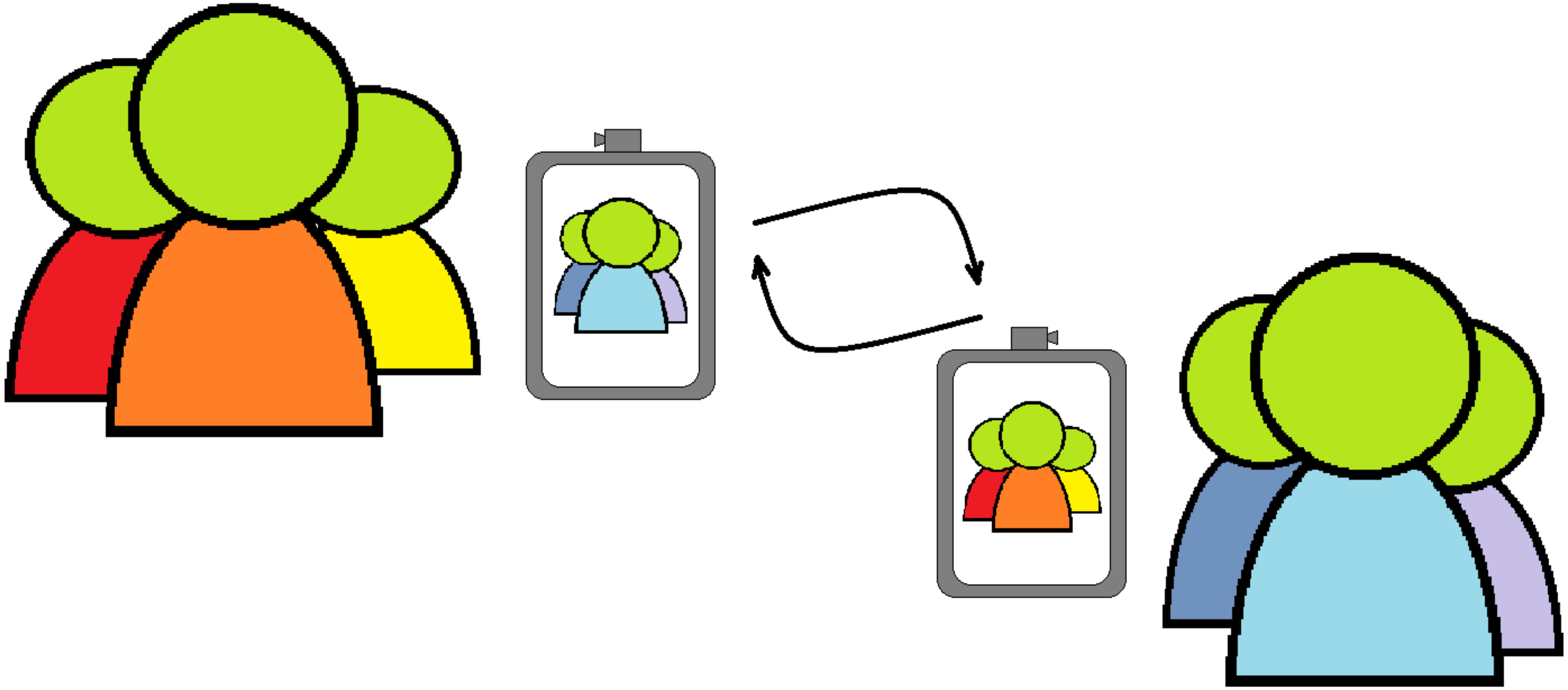
- a step towards smartphone-appropriate memory scheduler designs
- trace-based methodology
  - memory traces with dependence information
- software-based methodology to approximate hardware accelerator behavior
- we study:
  - address mapping schemes
  - memory schedulers
  - Video Conference Workload
  - other smartphone workloads

Content

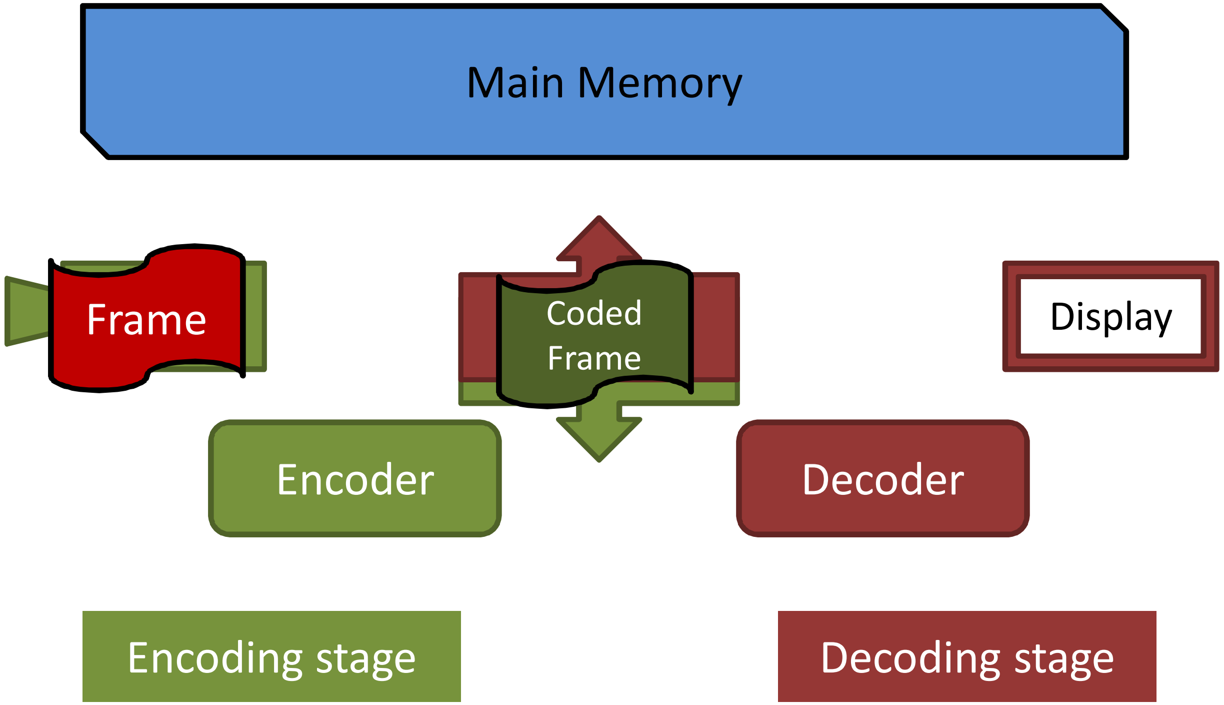
1. Video Conference
2. Modeling Specialised Hardware
3. Infrastructure Overview
4. Results
   - Address Mapping Scheme
   - Scheduler Comparisons
5. Summary

Video Conference

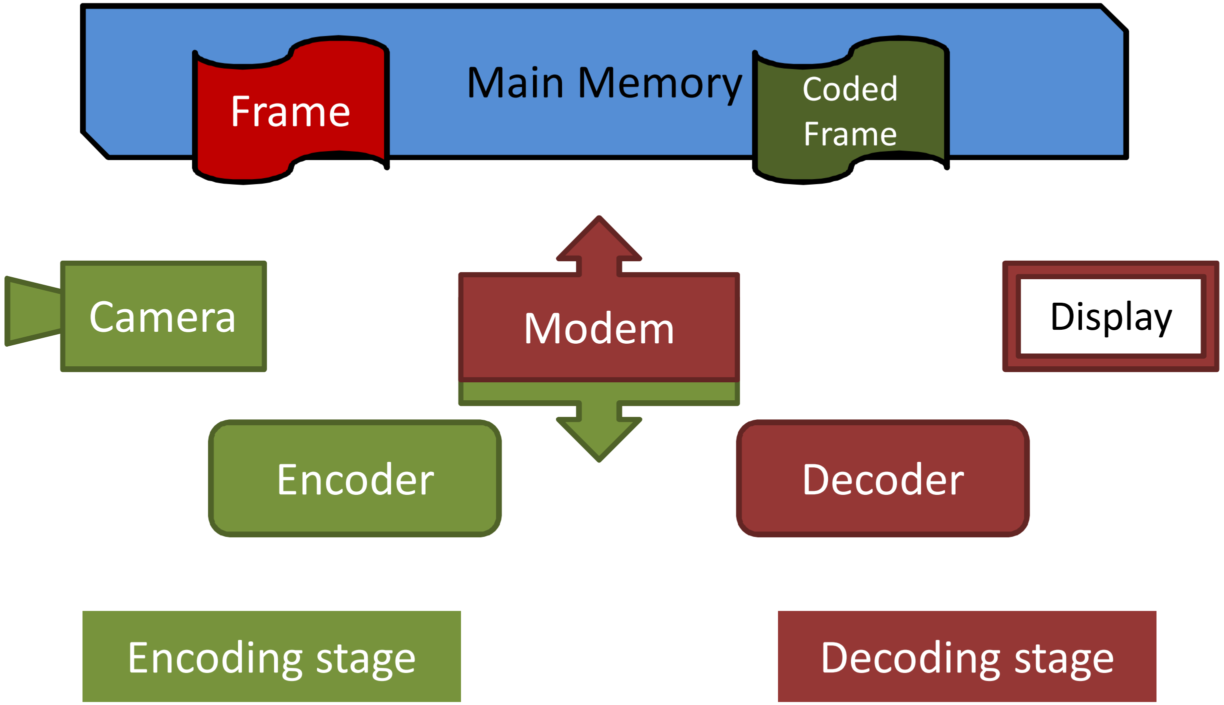
- two-way video call
  - a conversation between 2 persons, a video conference between meeting rooms, etc.



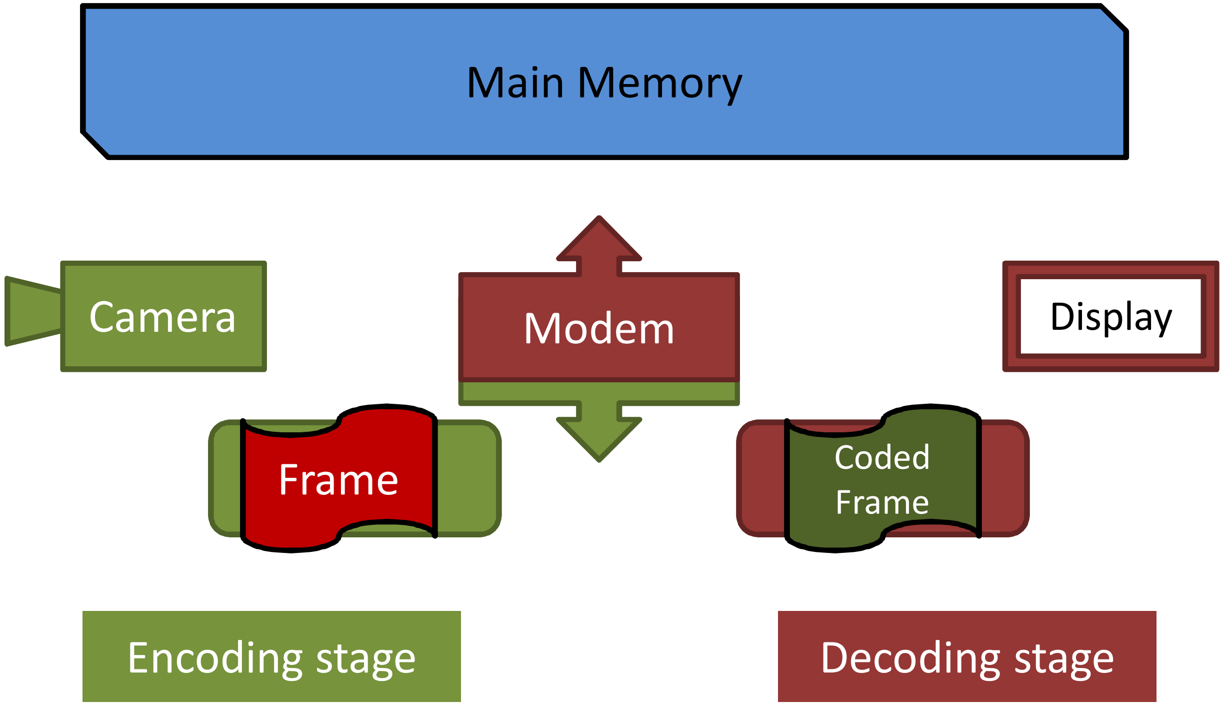
Video Conference Information Flow



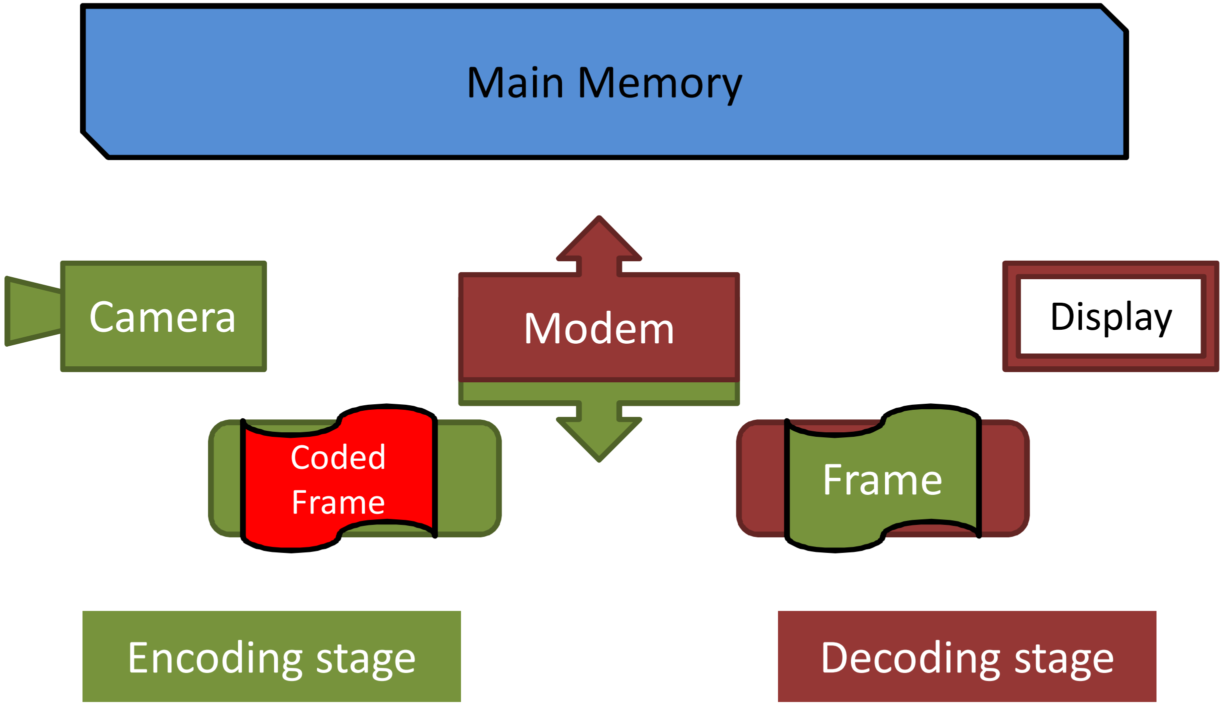
Video Conference Information Flow



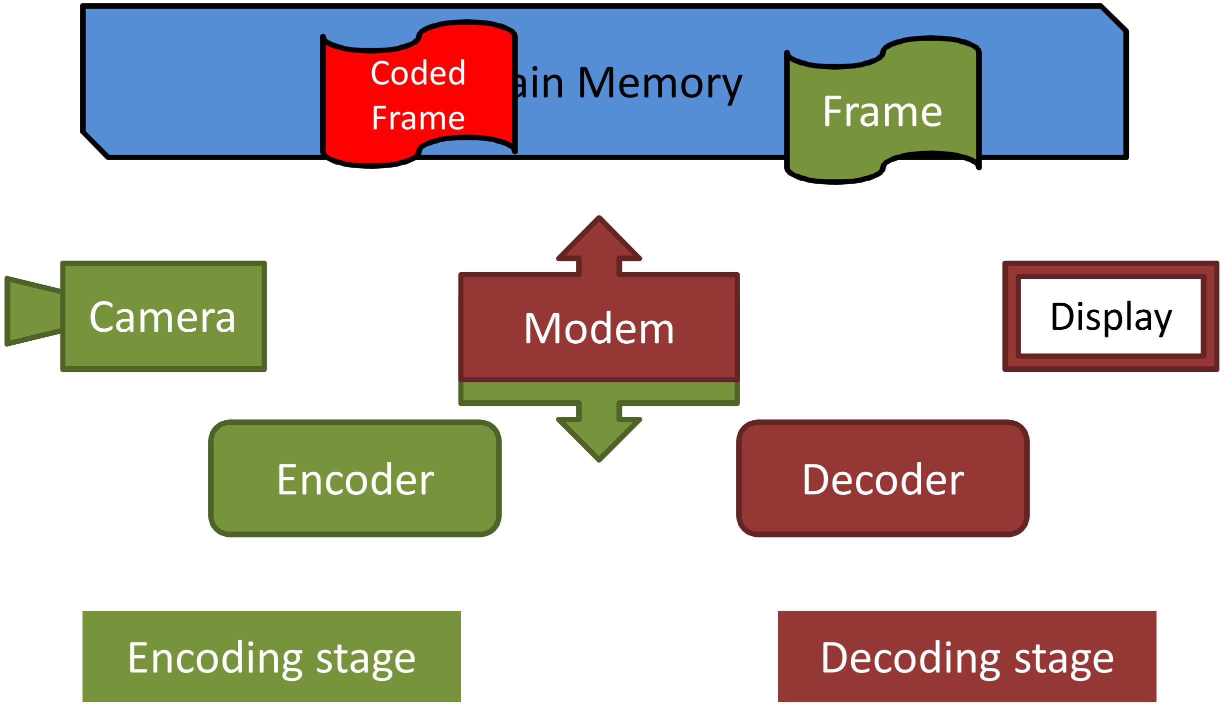
Video Conference Information Flow



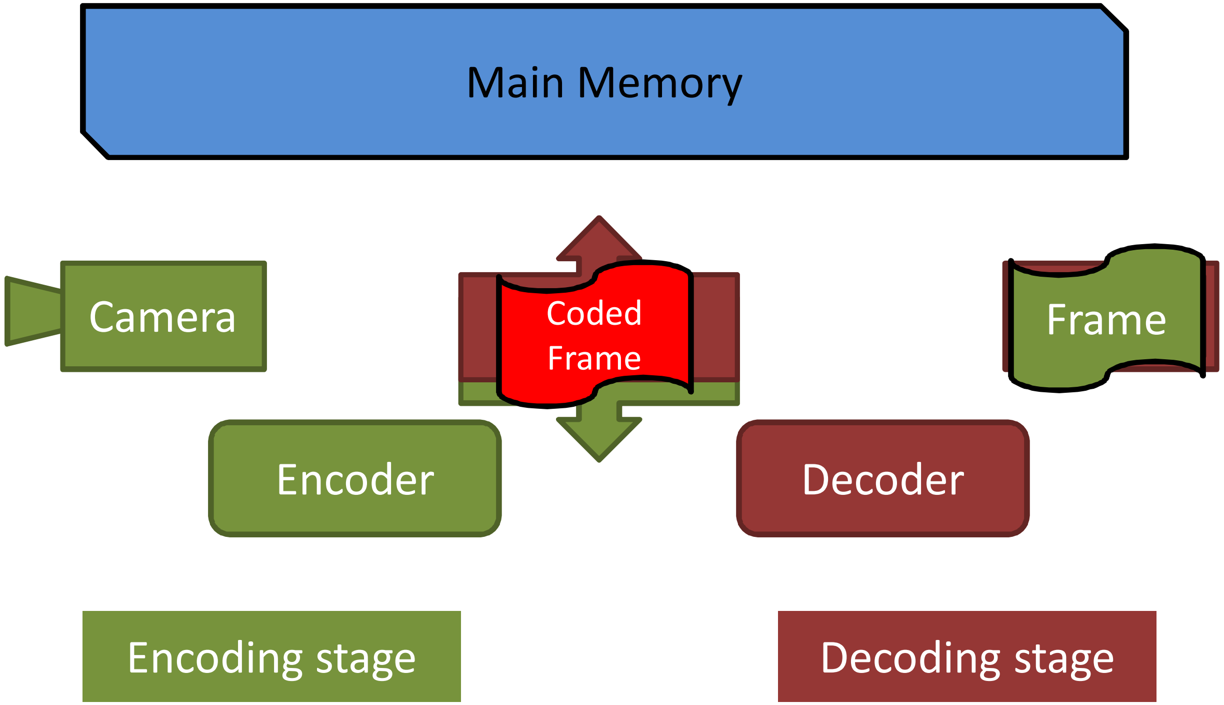
Video Conference Information Flow



Video Conference Information Flow



Video Conference Information Flow



Main Memory

Camera

Coded Frame

Frame

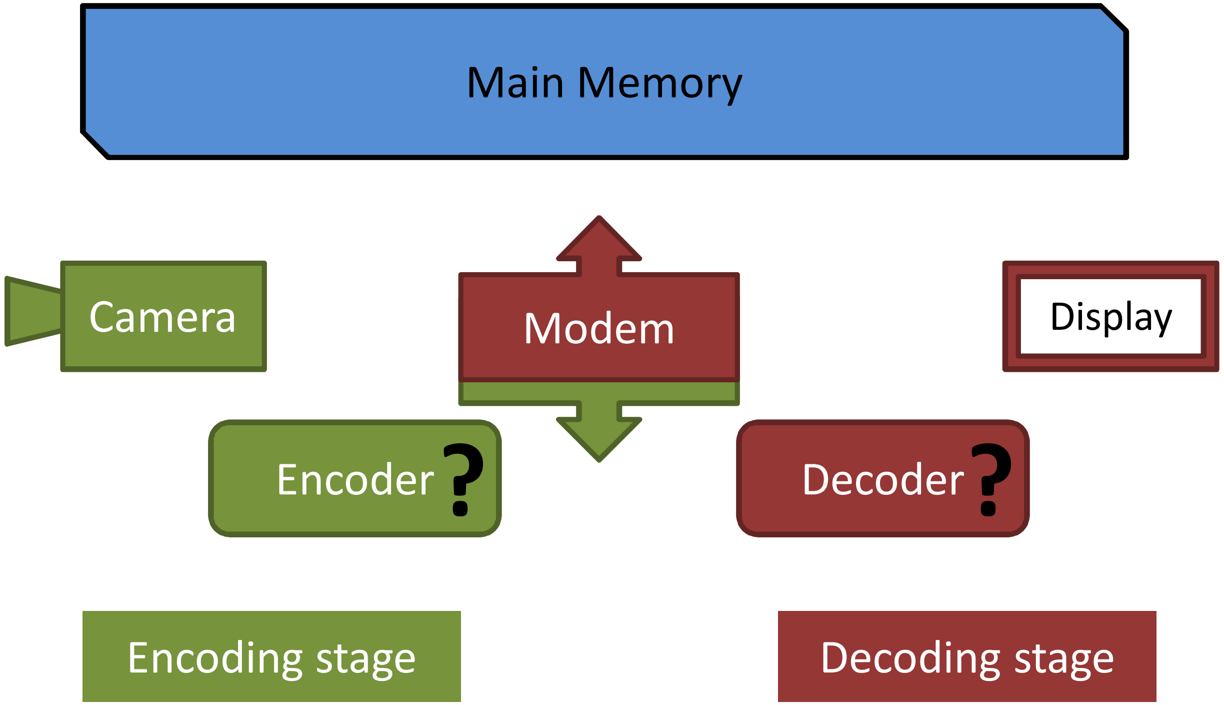
Encoder

Decoder

Encoding stage

Decoding stage

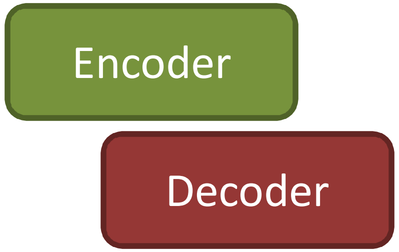
Video Conference Information Flow

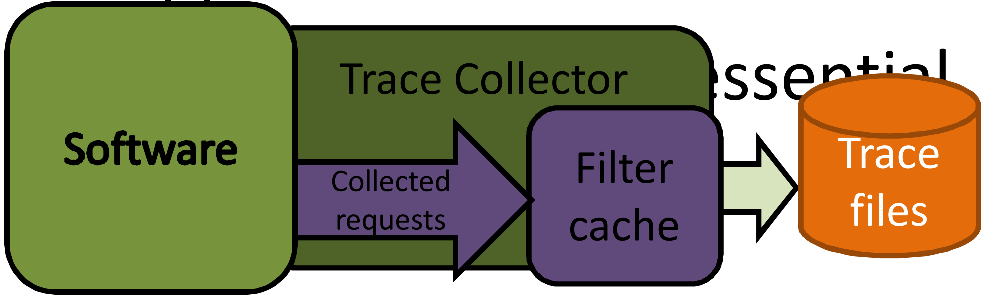


But how to simulate the encoder and decoder?

Modelling Specialised Hardware

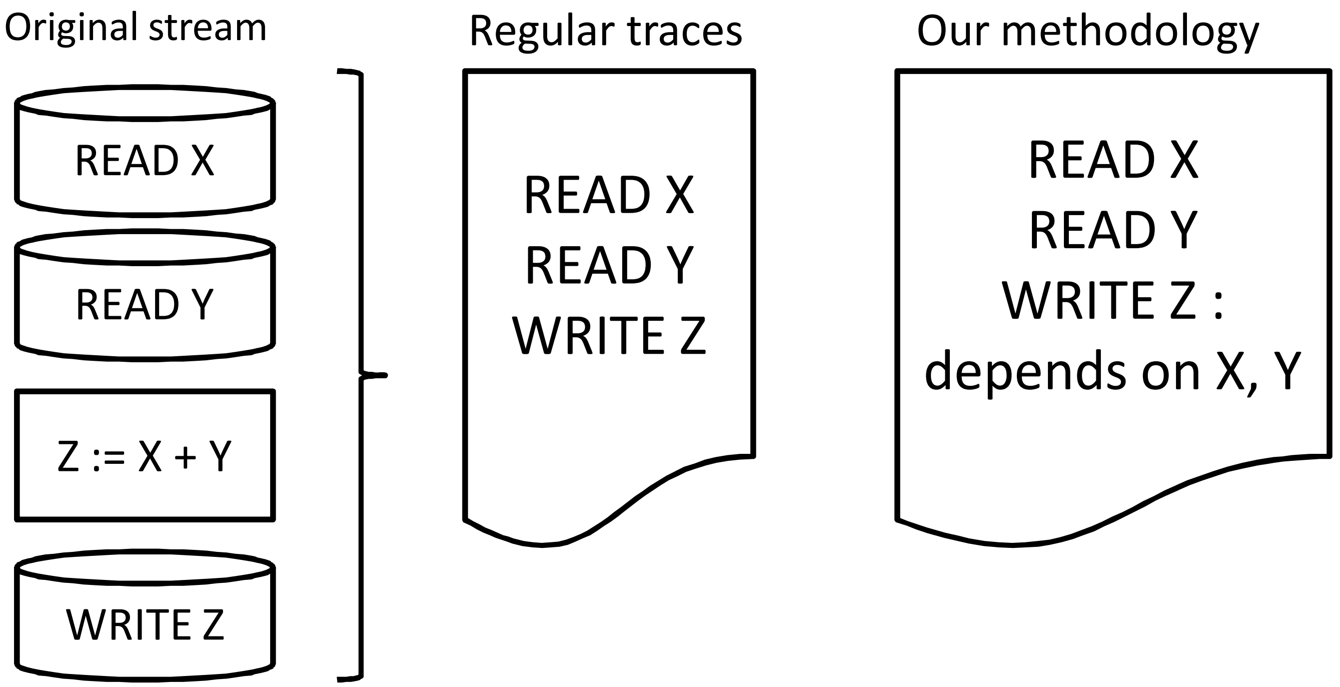
- devices designed for a specific use
  - optimised execution paths
  - buffers and small caches right where needed
- ideally: collect traffic from a real device
- our approach:
  - instrument software application and collect traces
  - use cache to filte ssential pattern



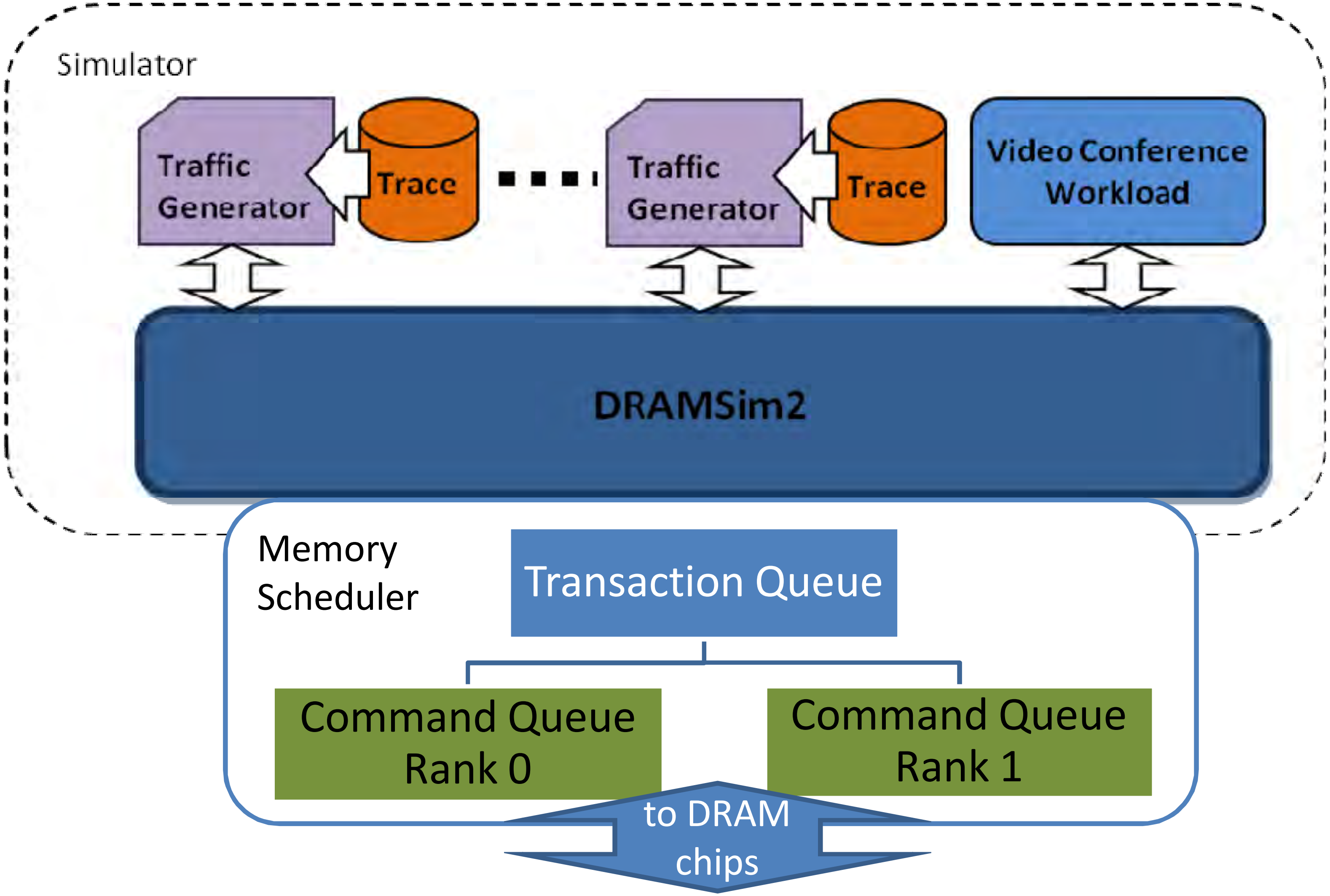


Maintaining Ordering Constraints

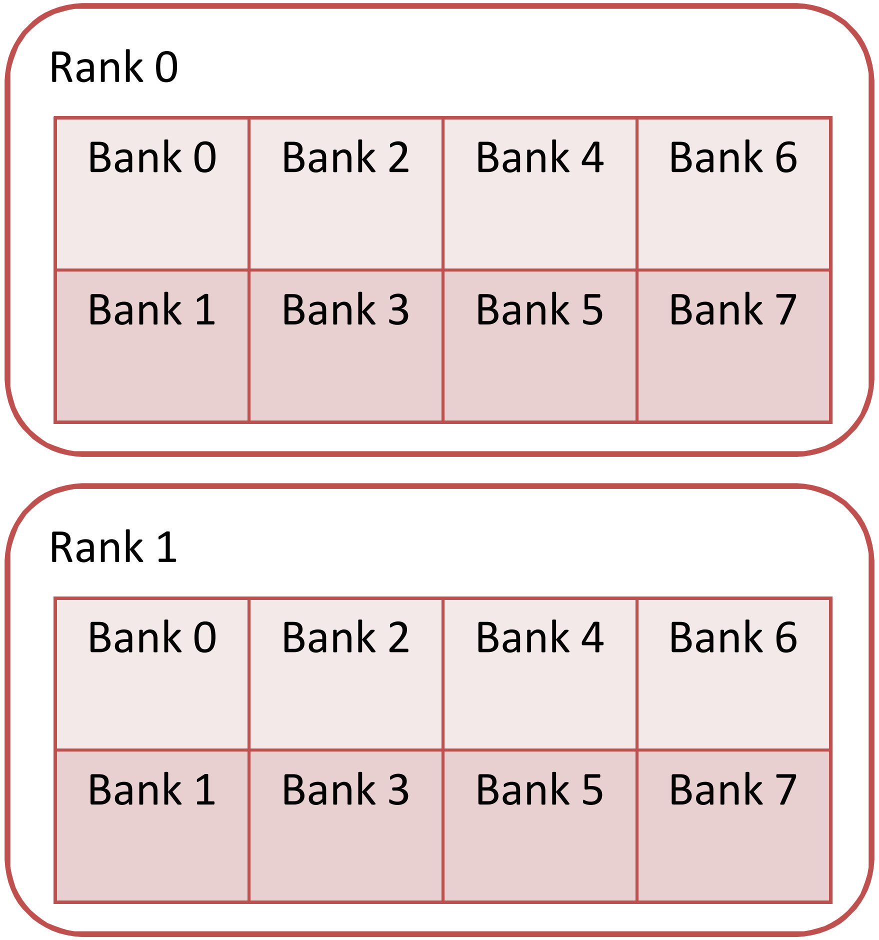
- traces -> no relationships between requests
- store dataflow information to limit requests



Infrastructure overview



DRAM Organisation



Rank 0

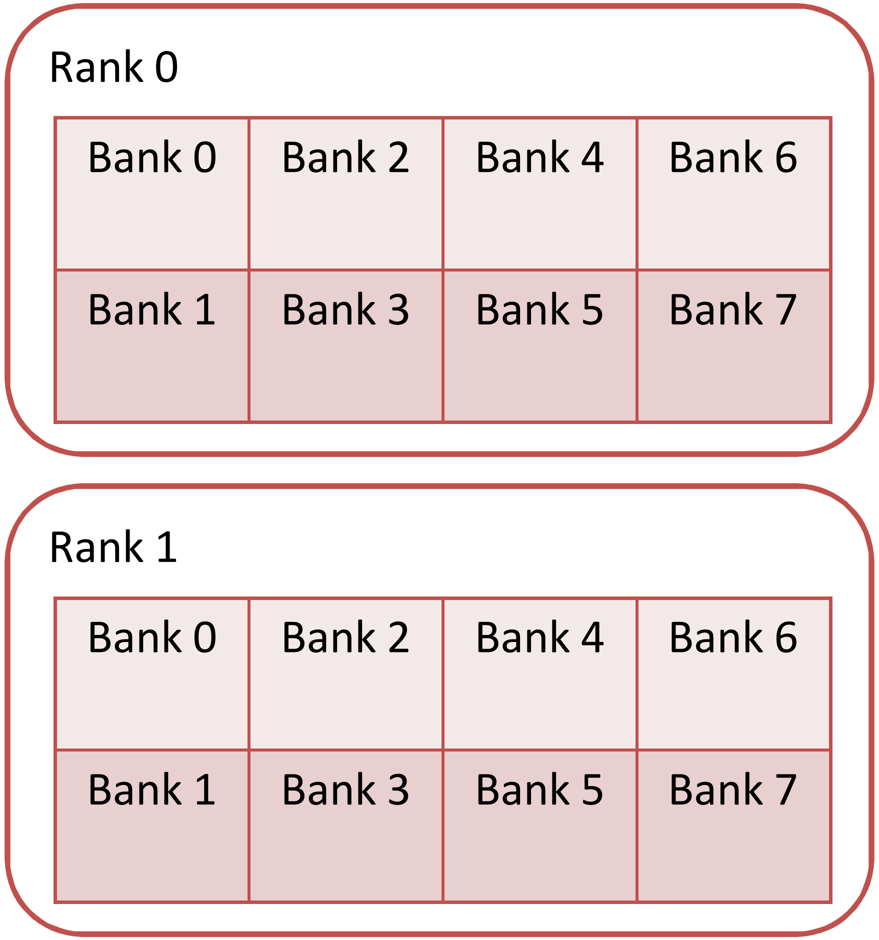
Bank 0	Bank 2	Bank 4	Bank 6
Bank 1	Bank 3	Bank 5	Bank 7

Rank 1

Bank 0	Bank 2	Bank 4	Bank 6
Bank 1	Bank 3	Bank 5	Bank 7

DRAM Organisation

- 4GB *DDR3 SDRAM* at 800 MHz
- logical organisation:
  - 2 *ranks* [1 bit]
  - 8 *banks* [3 bits]
  - 16384 *row* [14 bits]
  - 256 *columns* [8 bits]



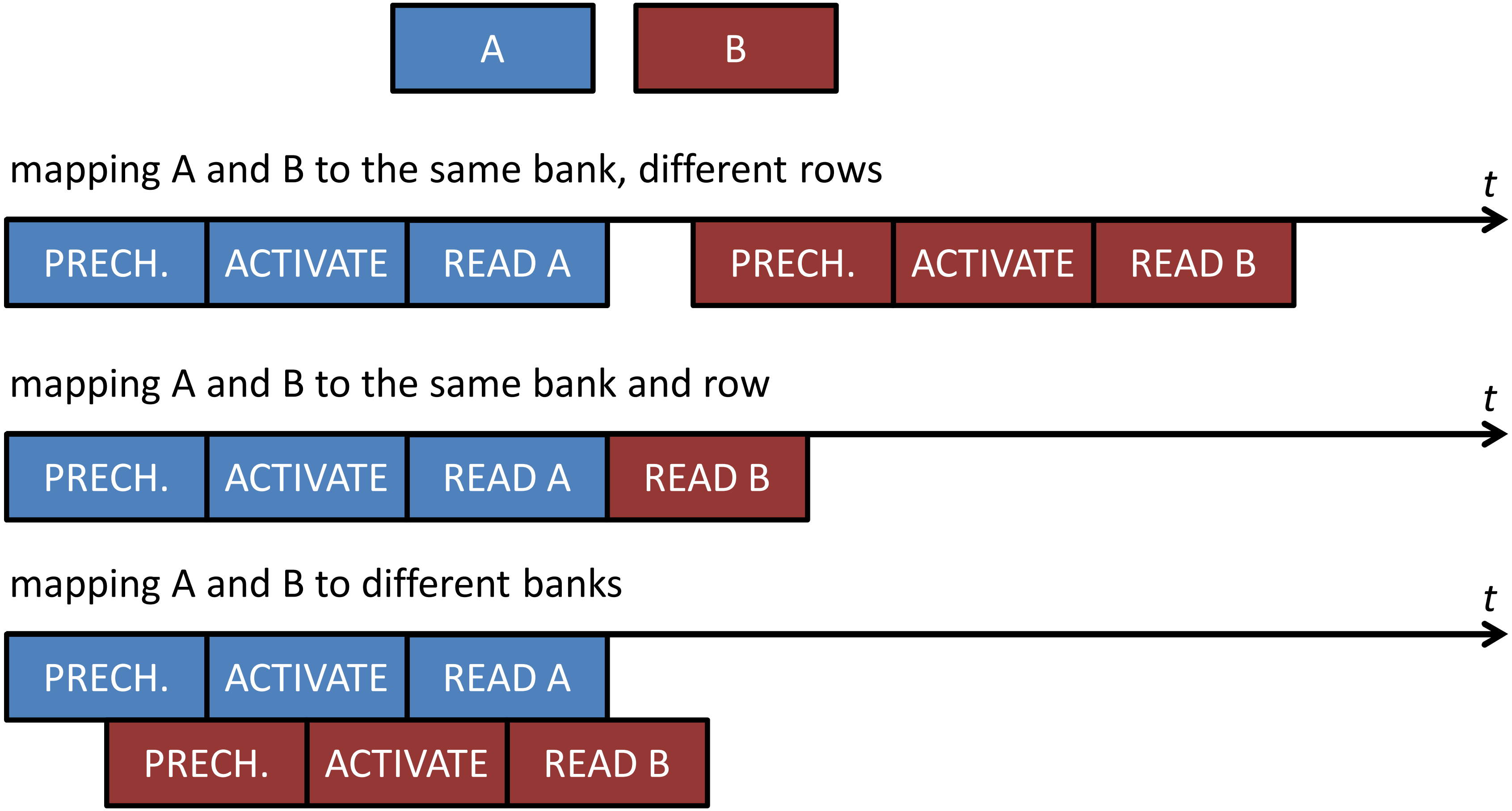
Rank 0

Bank 0	Bank 2	Bank 4	Bank 6
Bank 1	Bank 3	Bank 5	Bank 7

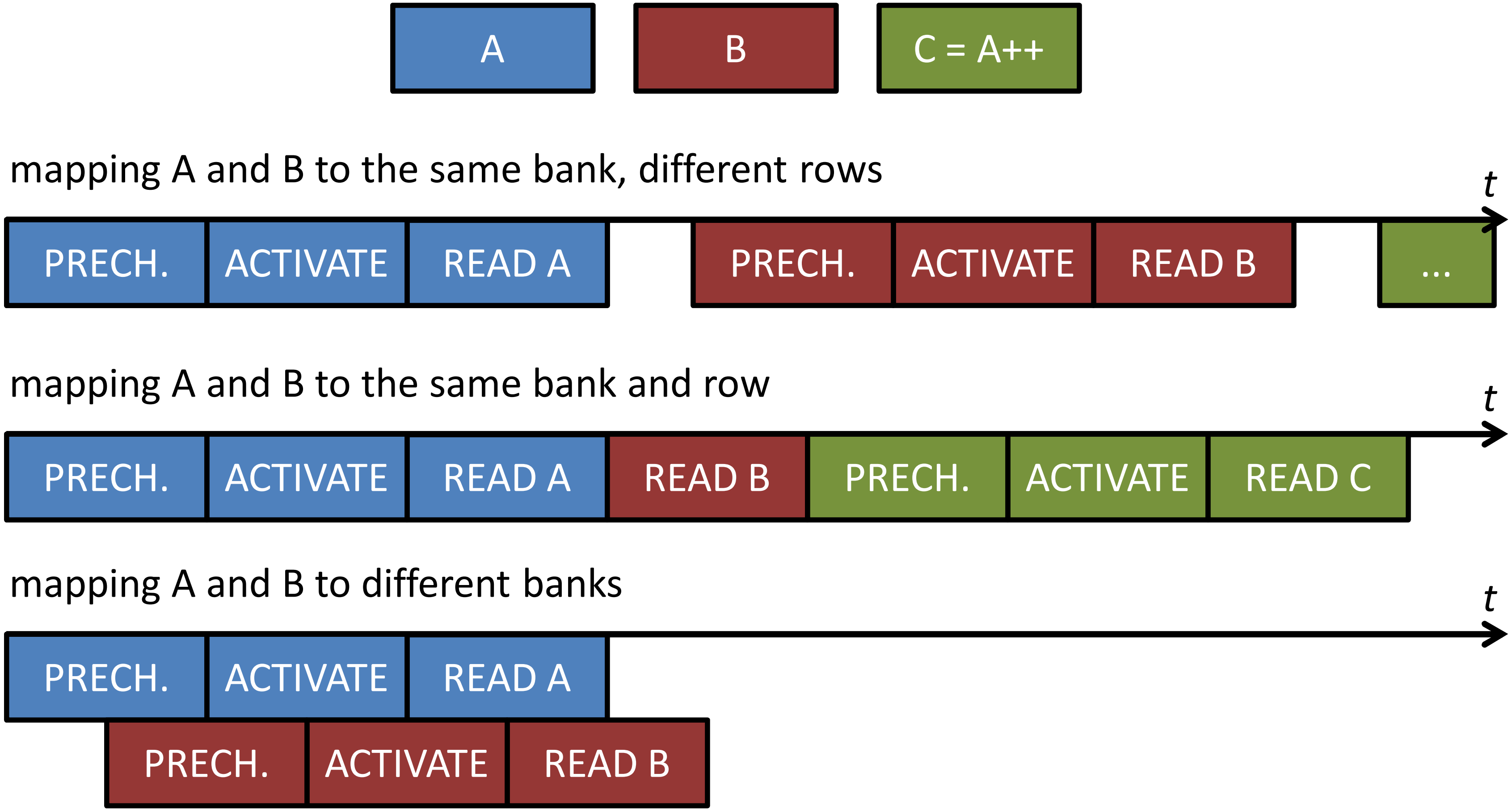
Rank 1

Bank 0	Bank 2	Bank 4	Bank 6
Bank 1	Bank 3	Bank 5	Bank 7

Requests, Location and Scheduling


A
B
mapping A and B to the same bank, different rows
t
PRECH.
ACTIVATE
READ A
PRECH.
ACTIVATE
READ B
mapping A and B to the same bank and row
t
PRECH.
ACTIVATE
READ A
READ B
mapping A and B to different banks
t
PRECH.
ACTIVATE
READ A
PRECH.
ACTIVATE
READ B

Requests, Location and Scheduling



A | B | C = A++

mapping A and B to the same bank, different rows

PRECH. | ACTIVATE | READ A | PRECH. | ACTIVATE | READ B | ...

t

mapping A and B to the same bank and row

PRECH. | ACTIVATE | READ A | READ B | PRECH. | ACTIVATE | READ C

t

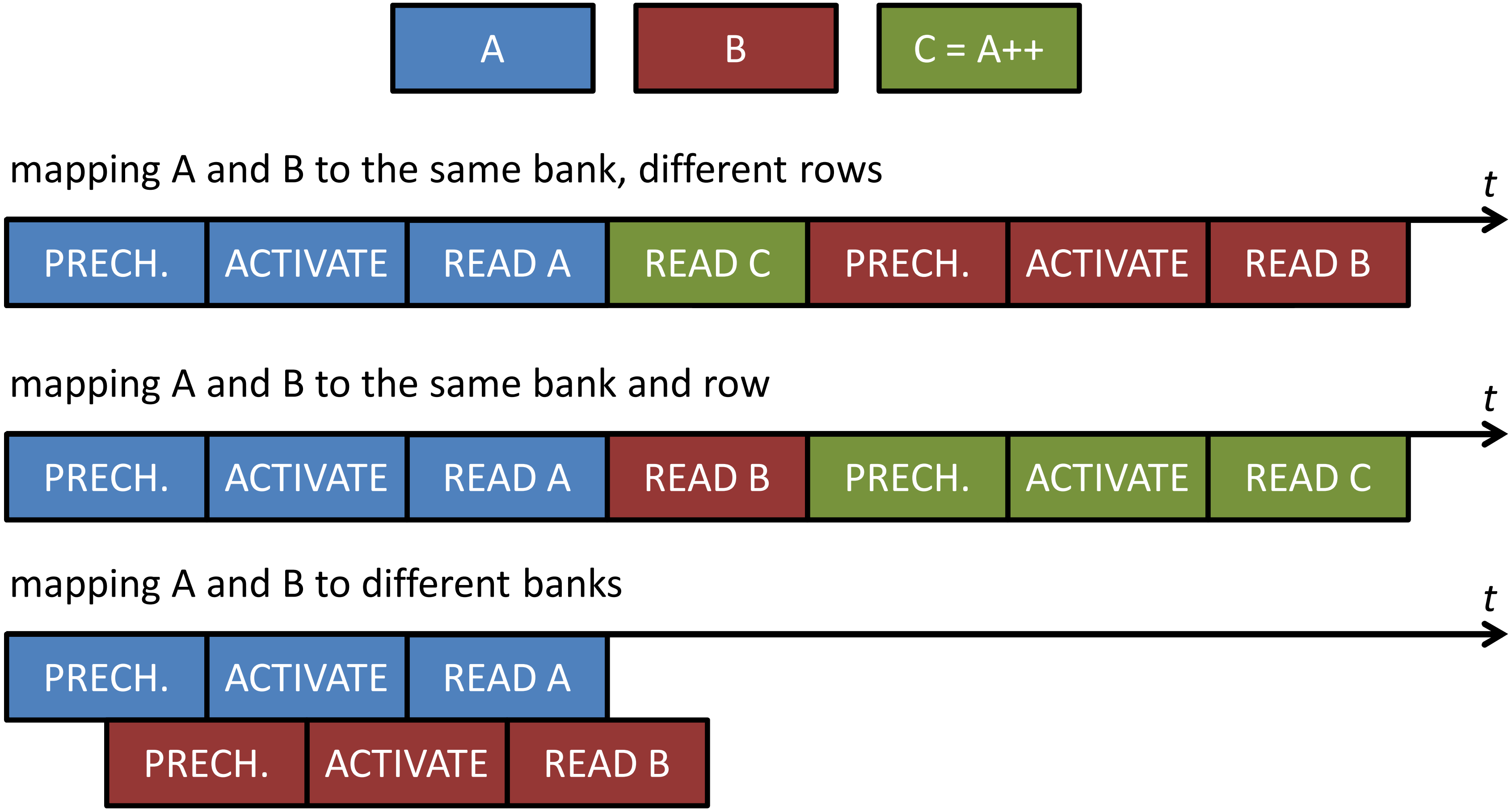
mapping A and B to different banks

PRECH. | ACTIVATE | READ A

PRECH. | ACTIVATE | READ B

t

Requests, Location and Scheduling



Address mapping and scheduling can have great effect!

Content

1. Video Conference
2. Modeling Specialised Hardware
3. Infrastructure Overview
4. Results
   - Address Mapping Scheme
   - Scheduler Comparisons
5. Summary

Content

1. Video Conference
2. Modeling Specialised Hardware
3. Infrastructure Overview
4. Results
   - Address Mapping Scheme
   - Scheduler Comparisons
5. Summary

Test Configurations

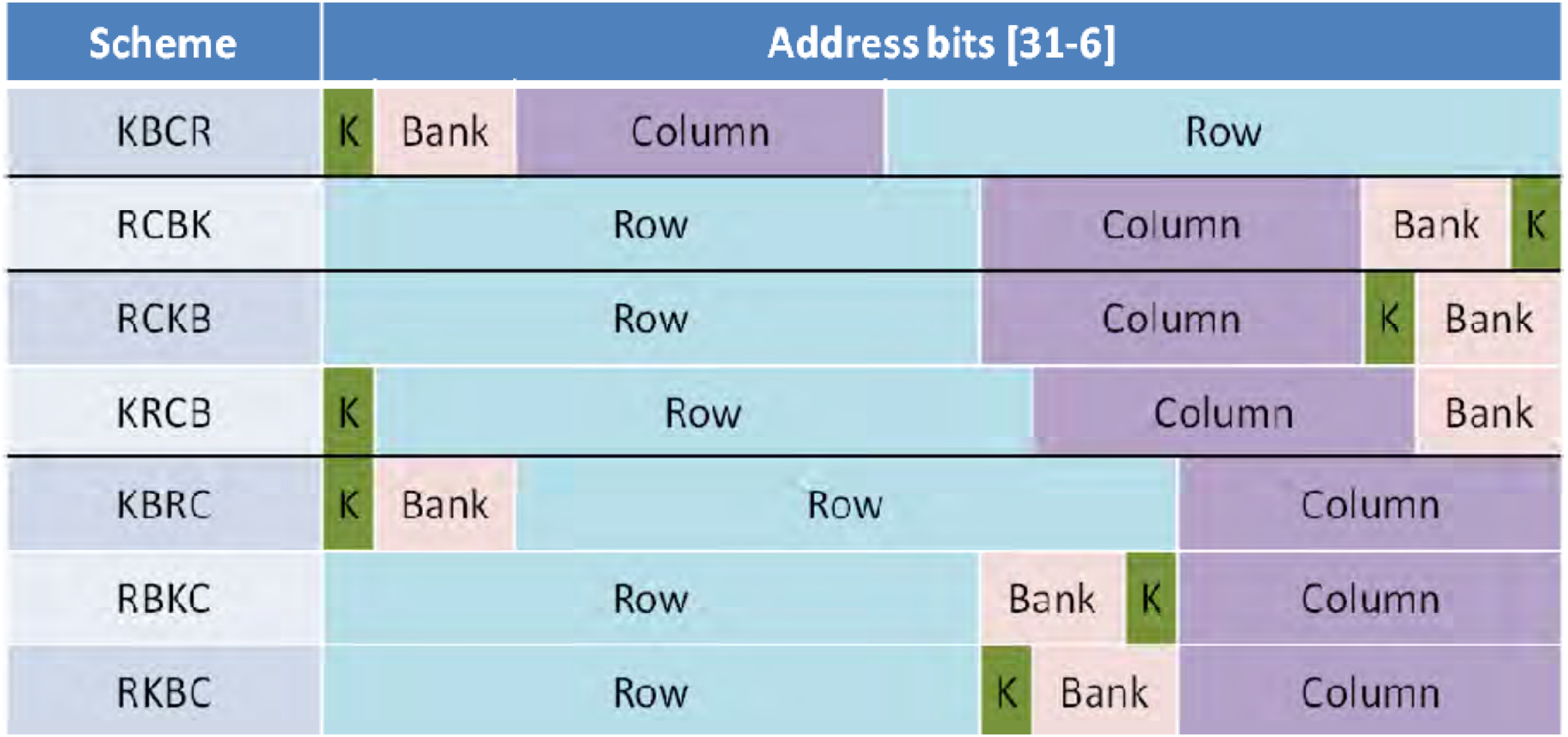
- scheduler configuration:

	Max Buffer Hits	Command Queue	Transaction Queue	Write Queue	High Mark	Low Mark
Limited	32	8	24	16	12	8
Maximum	1024	512	512	64	60	50

* Our implementation of *TFRR* does not use TQ, so we increase CQ to 20 and 1024

- we examine:
  - address mapping on the VCW workload
  - schedulers on a combination of:
    - Web Browsing, Face Detection, VCW

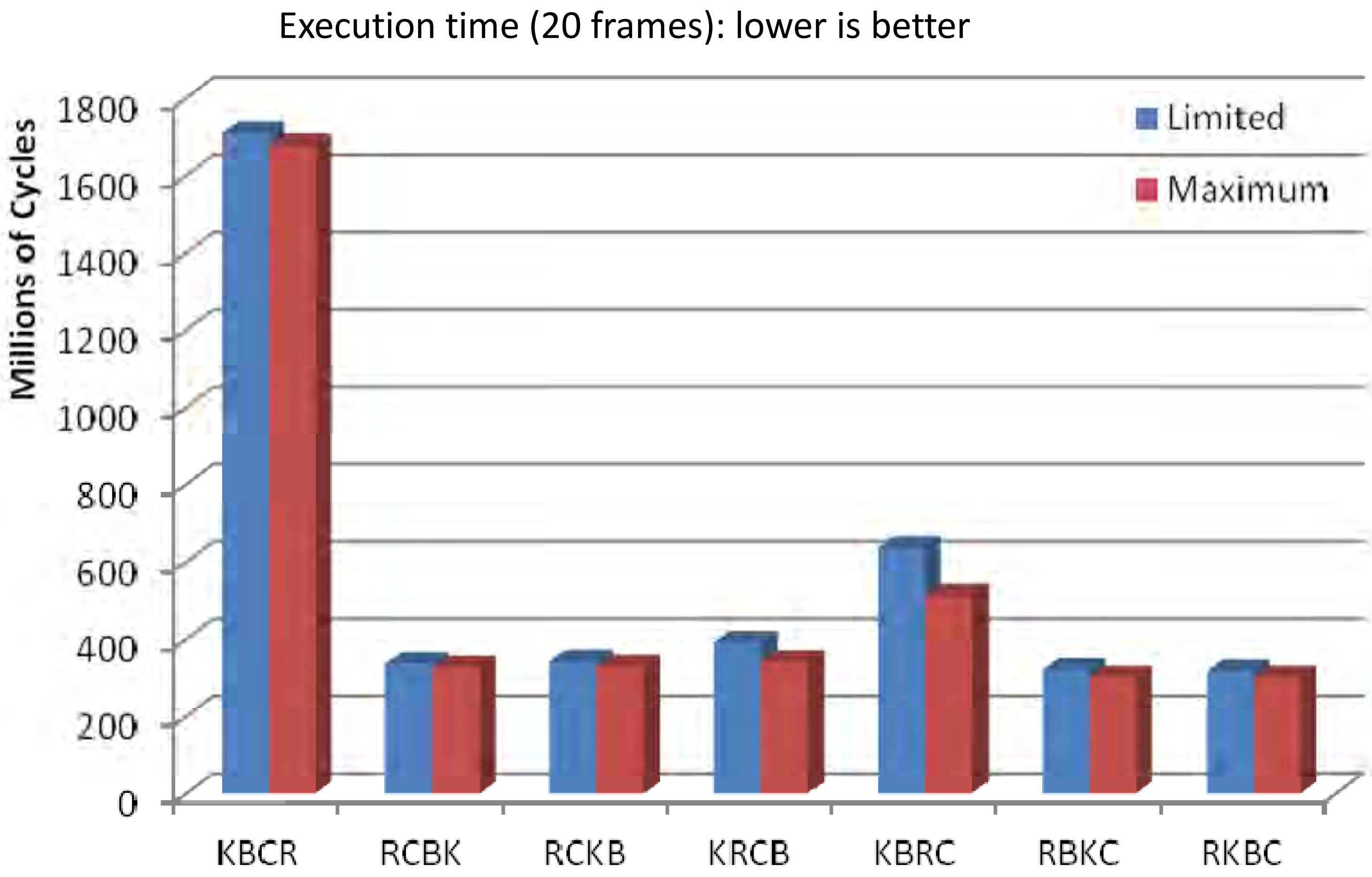
Address Mapping Schemes



Scheme	Address bits [31-6]			
KBCR	K	Bank	Column	Row
RCBK	Row	Column	Bank	K
RCKB	Row	Column	K	Bank
KRCB	K	Row	Column	Bank
KBRC	K	Bank	Row	Column
RBKC	Row	Bank	K	Column
RKBC	Row	K	Bank	Column

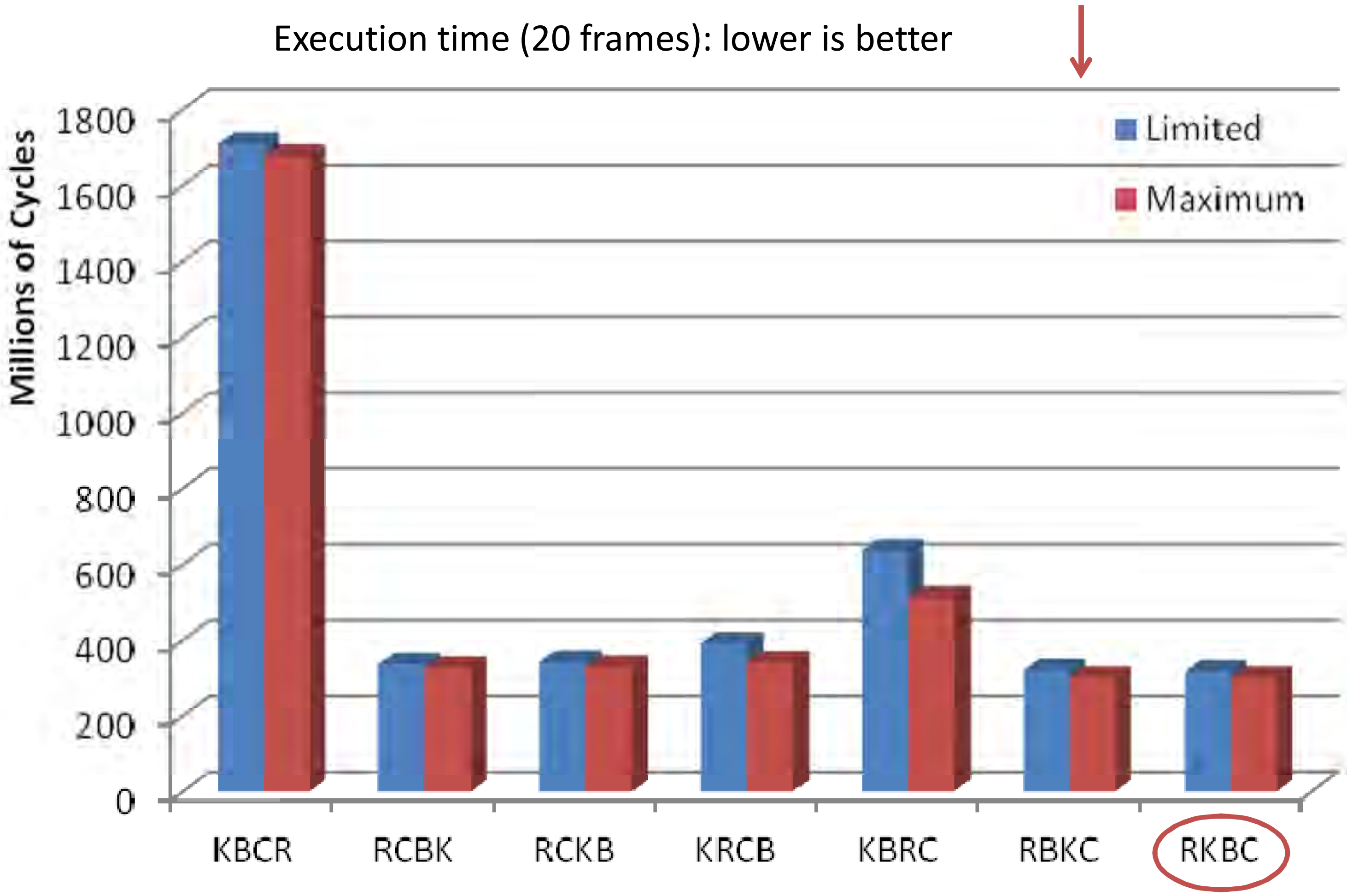
K designates the rank bits

Address Mapping Schemes: Results



Address Mapping Schemes: Results

Execution time (20 frames): lower is better



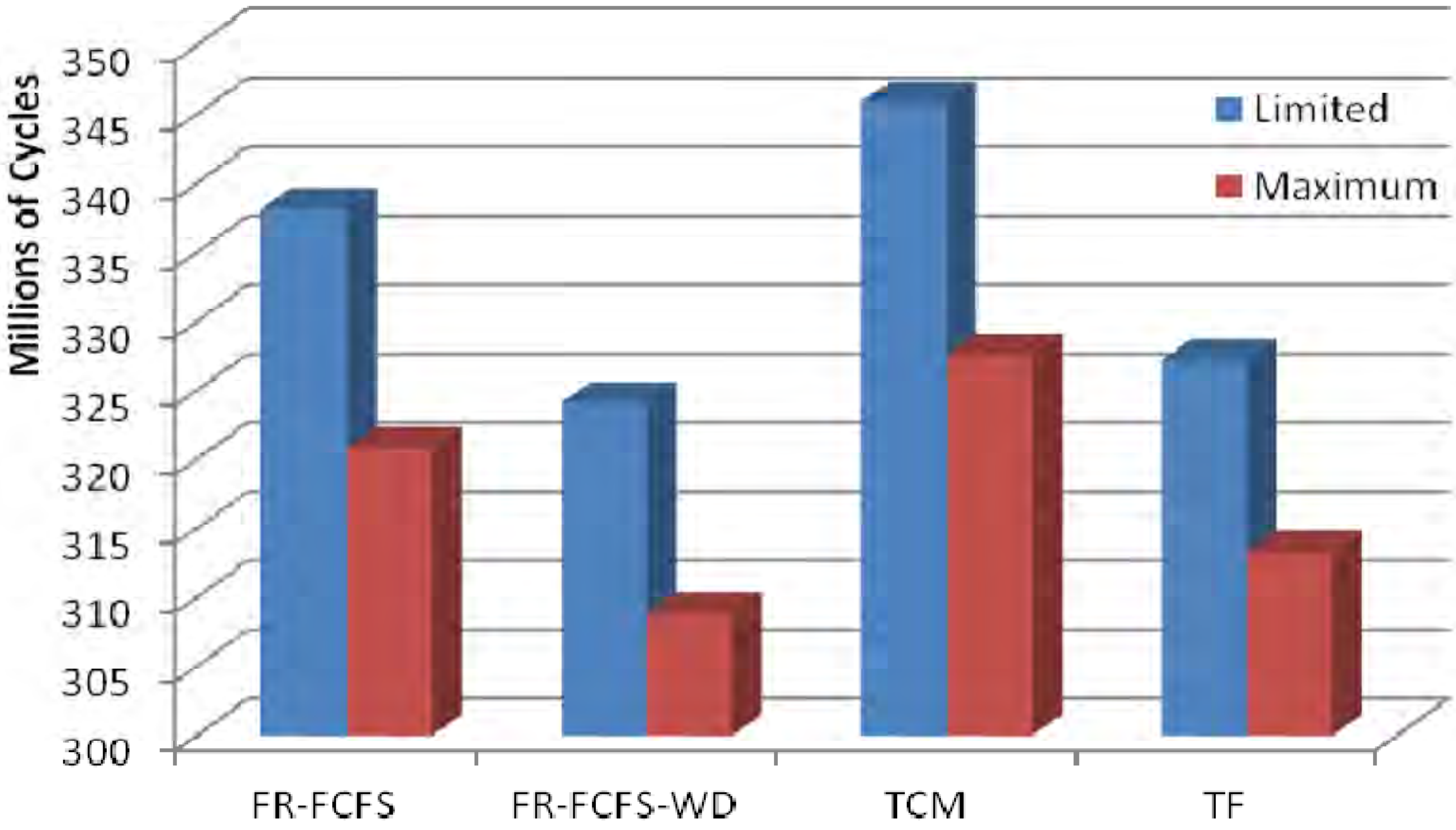
RKBC performs best due to high row buffer locality combined with good parallelism

Schedulers

- *First Ready – First Come, First Served* (*FR-FCFS*)
  - baseline, simple
- *FR-FCFS* with *Write Drain* (*FR-FCFS-WD*)
  - delays *WRITEs* to improve *READ* latency
- *Thread-Fair* memory scheduler (*TF*)
  - prioritises requests from *ROB* head
- *Thread Clustering* memory scheduler (*TCM*)
  - thread-ranking strict prioritisation

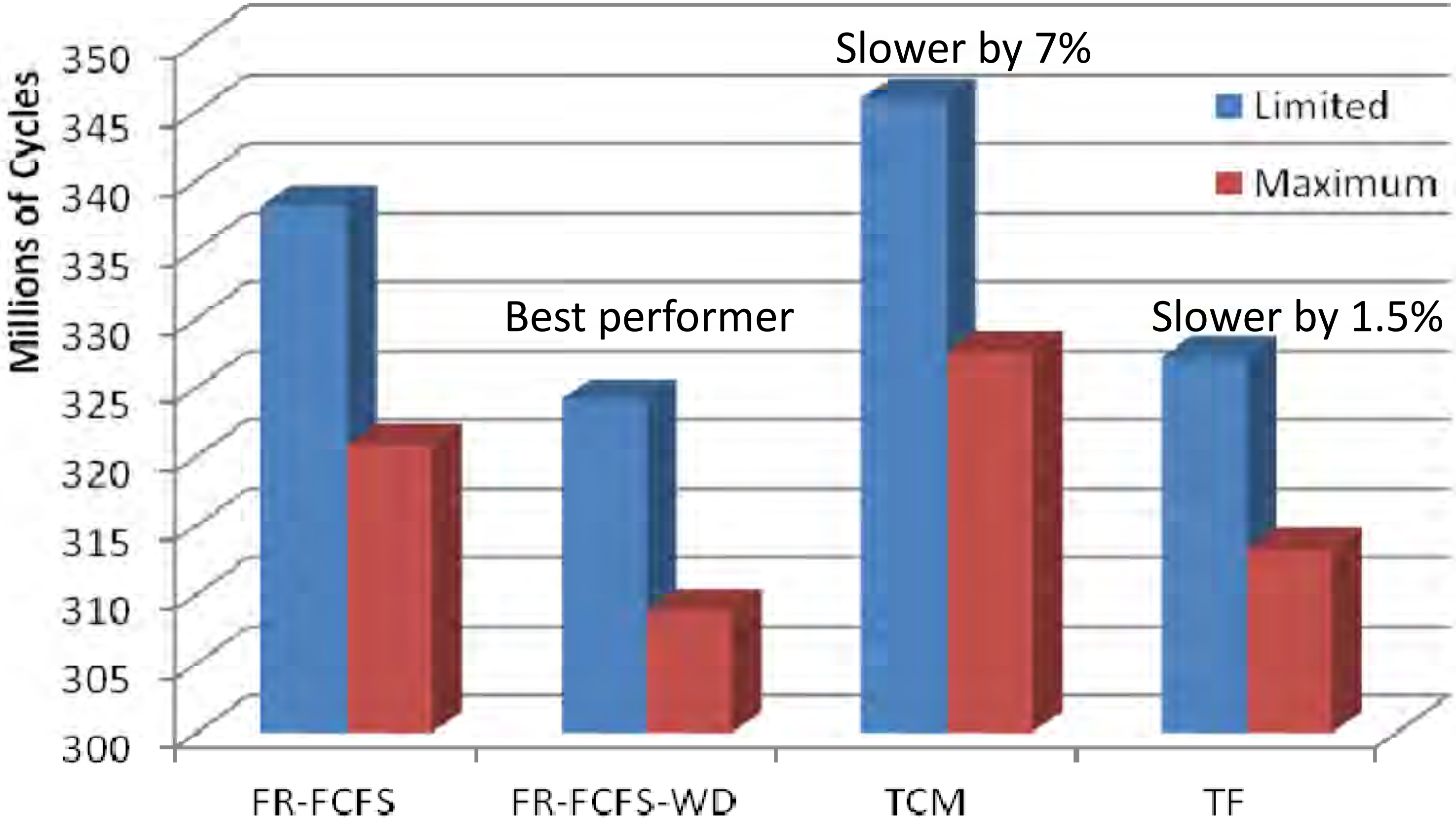
Scheduler Comparison

Execution time: lower is better



Scheduler Comparison

Execution time: lower is better



Slower by 7%

Best performer

Slower by 1.5%

Summary

- our contributions
  - trace-based methodology
    - request issuing limited by dataflow
    - uses cache to model specialised hardware
      - no validation (yet)
  - Video Conference Workload
    - model typical smartphone usage
    - our tests show it is memory bound

Summary: Our Findings

- address mapping has significant impact
  - best scheme runs in 1/5 time of the worst one
- compare schedulers
  - older, simpler: *FR-FCFS*, *FR-FCFS-WD*
  - newer, thread-concious: *Thread-Fair*, *Thread Clustering*
  - found that simpler perform better
  - *Write Drain* mode useful

THANK YOU FOR YOUR ATTENTION!