VIEW GITHUB REPOSITORY

OpenDRAM: A Modular, High-performance Soft Memory Controller for DDR4 DRAM

Ali Abbasi †, Danesh Germchi †,
Amin Katani, Mohamed Hassan, Rodolfo Pellizzoni

† Equal contribution

FPT'25 Shanghai

International Conference on Field Programmable Technology (FPT2025)
2-5th December, Shanghai, China

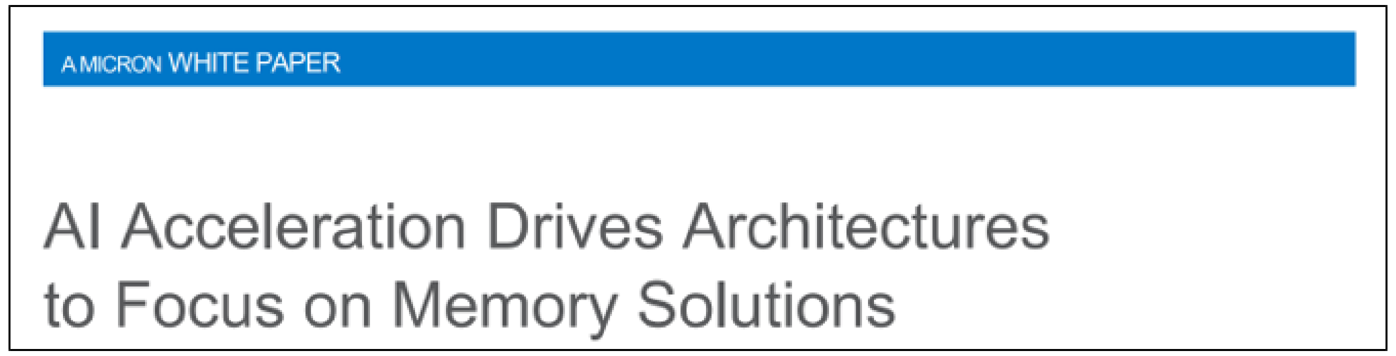
UNIVERSITY OF WATERLOO

McMaster University

Motivation

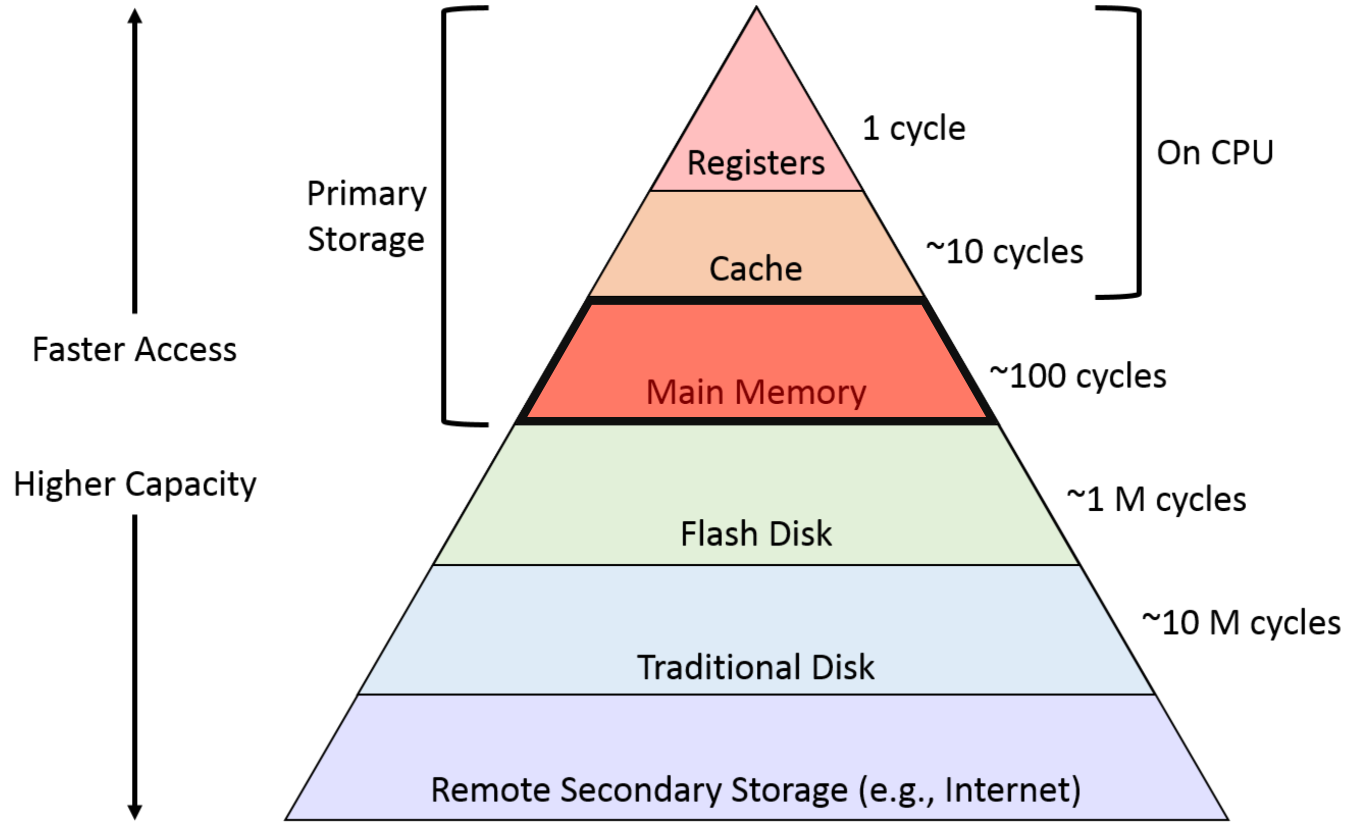
VIEW GITHUB REPOSITORY

- Modern workloads require massive memory capacity and bandwidth
- DRAMs are the preferred choice for main memory
- Memory controllers are the governors of DRAMs
- Need for an accessible open-source DRAM controller for research and evaluations
- Architectural simulators overlook hardware complexities
- Limited RTL implementations. Existing ones are either:
  - Not extensible/modular
  - Not high-performance
  - Not open-source



A MICRON WHITE PAPER

AI Acceleration Drives Architectures to Focus on Memory Solutions



The Memory Hierarchy

OpenDRAM is the first open-source, extensible, modular, high-performance DDR4 memory controller.

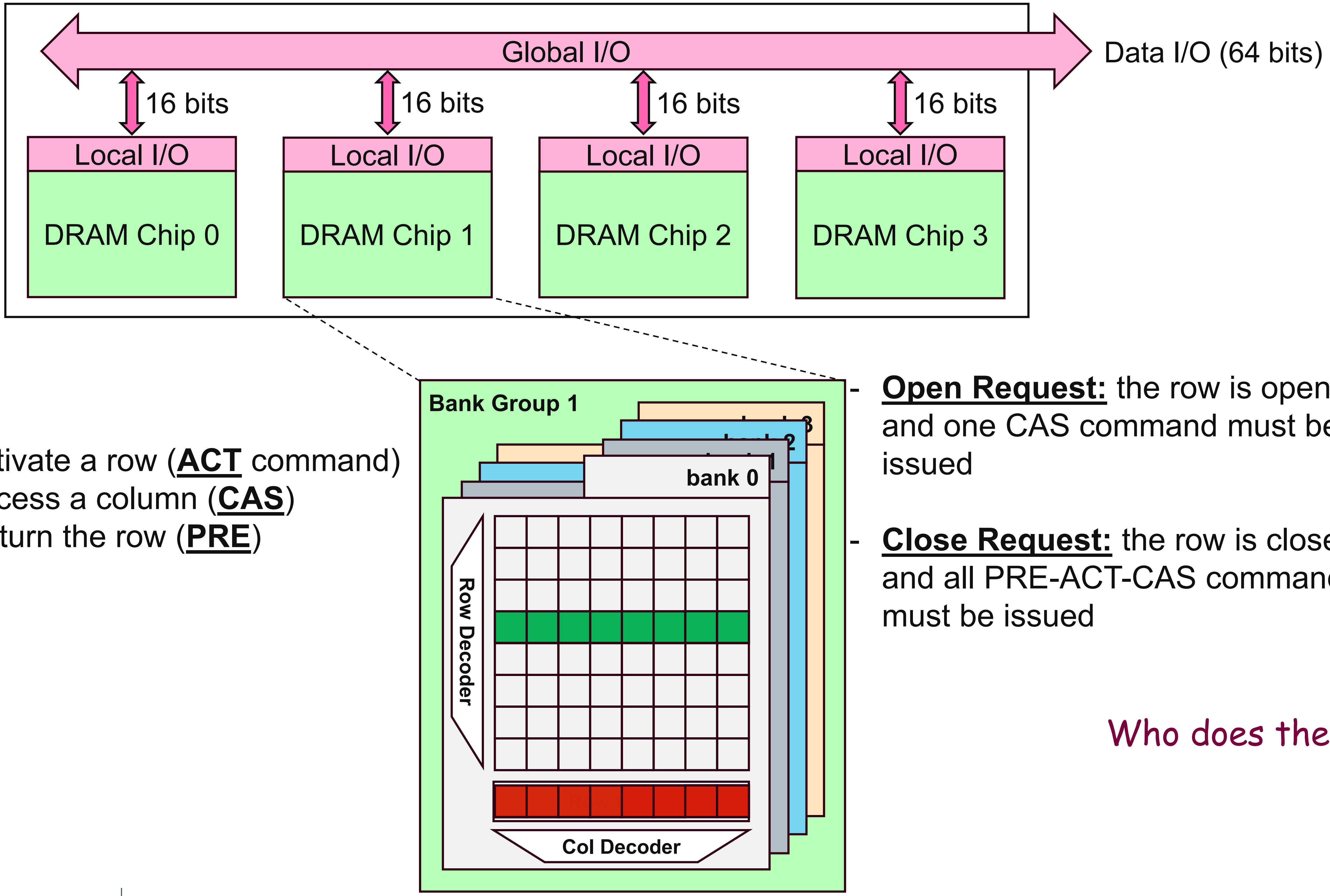
😎

Picture Source:
https://www.cs.swarthmore.edu/~kwebb/cs31/f18/memhierarchy/mem_hierarchy.html

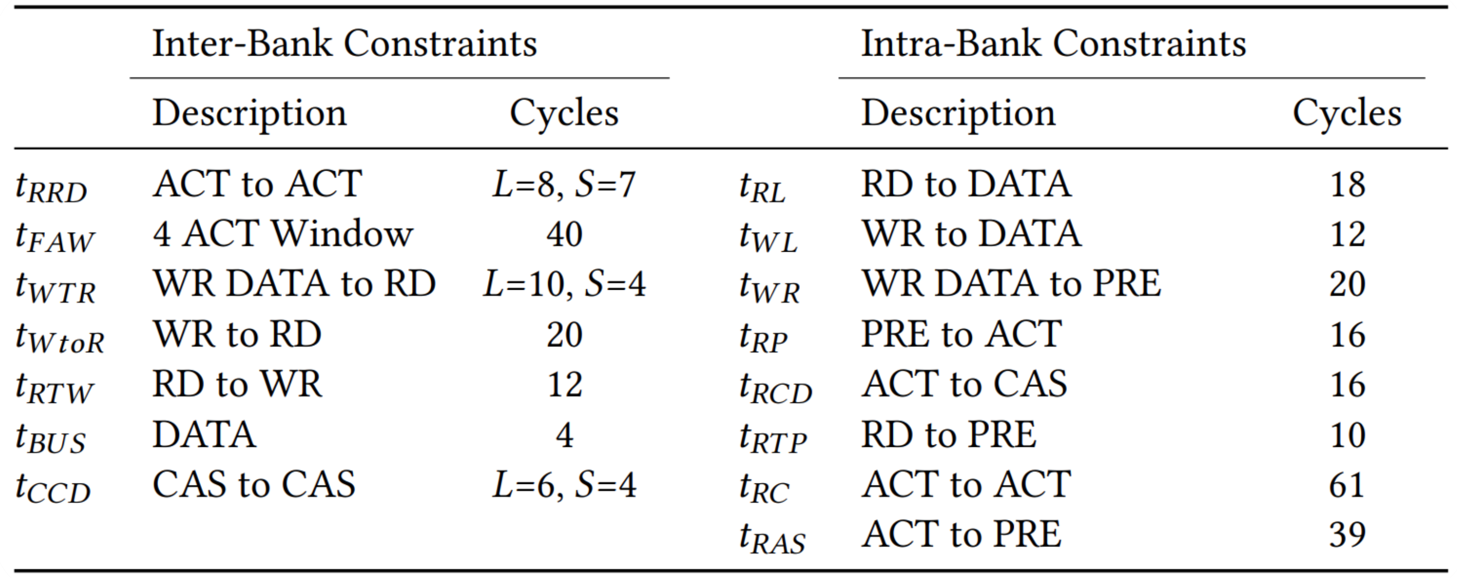
Background

DDR SDRAM - Organization

VIEW GITHUB REPOSITORY



1) Activate a row (**ACT** command)
2) Access a column (**CAS**)
3) Return the row (**PRE**)



	Inter-Bank Constraints			Intra-Bank Constraints	
	Description	Cycles		Description	Cycles
t_{RRD}	ACT to ACT	L=8, S=7	t_{RL}	RD to DATA	18
t_{FAW}	4 ACT Window	40	t_{WL}	WR to DATA	12
t_{WTR}	WR DATA to RD	L=10, S=4	t_{WR}	WR DATA to PRE	20
t_{WtoR}	WR to RD	20	t_{RP}	PRE to ACT	16
t_{RTW}	RD to WR	12	t_{RCD}	ACT to CAS	16
t_{BUS}	DATA	4	t_{RTP}	RD to PRE	10
t_{CCD}	CAS to CAS	L=6, S=4	t_{RC}	ACT to ACT	61
			t_{RAS}	ACT to PRE	39

- **Open Request:** the row is open, and one CAS command must be issued
- **Close Request:** the row is closed, and all PRE-ACT-CAS commands must be issued

- Capacitors are used for storing data bits
  - Requires periodic charge **REFRESH**
- Designed simple to remain cheap
  - Requires external initialization and calibration

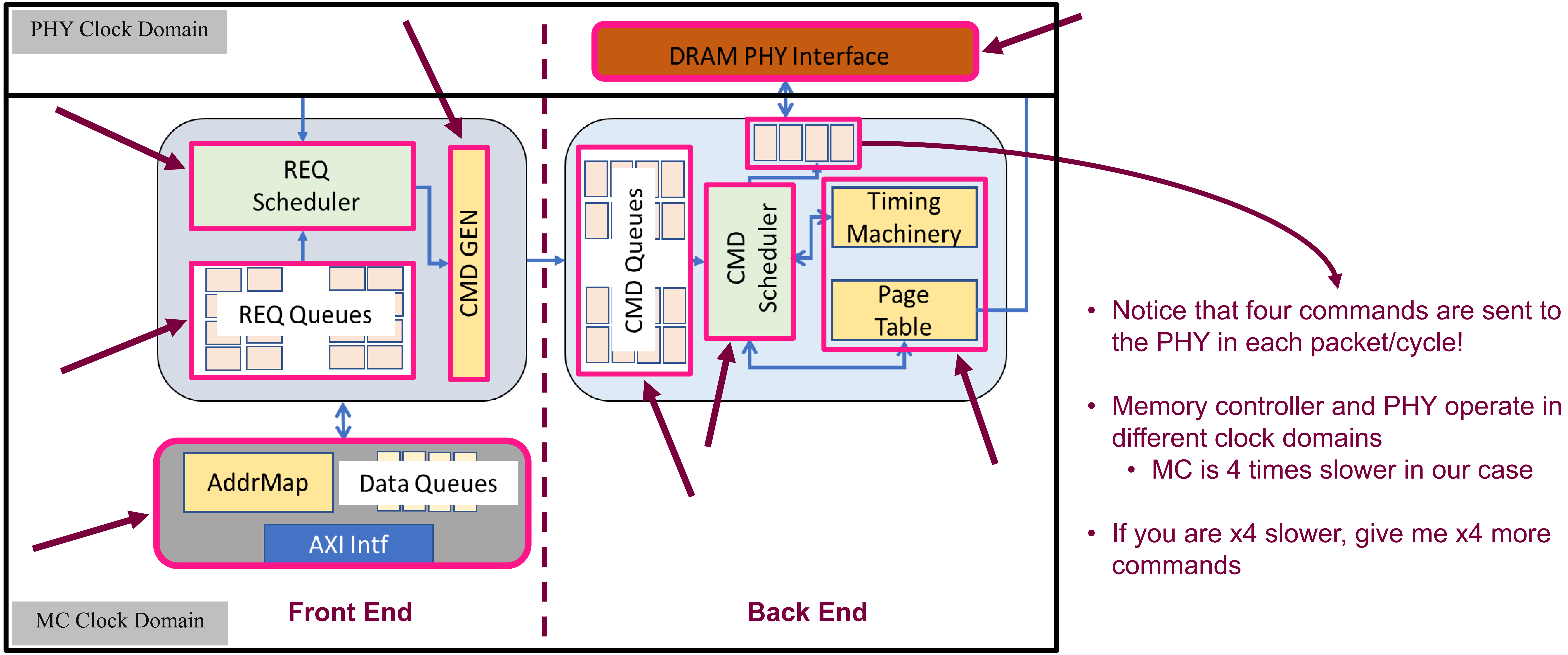
Who does the "smart work" so DRAM can stay cheap?

Memory Controller

Background

DDR SDRAM - Memory Controller

VIEW GITHUB REPOSITORY

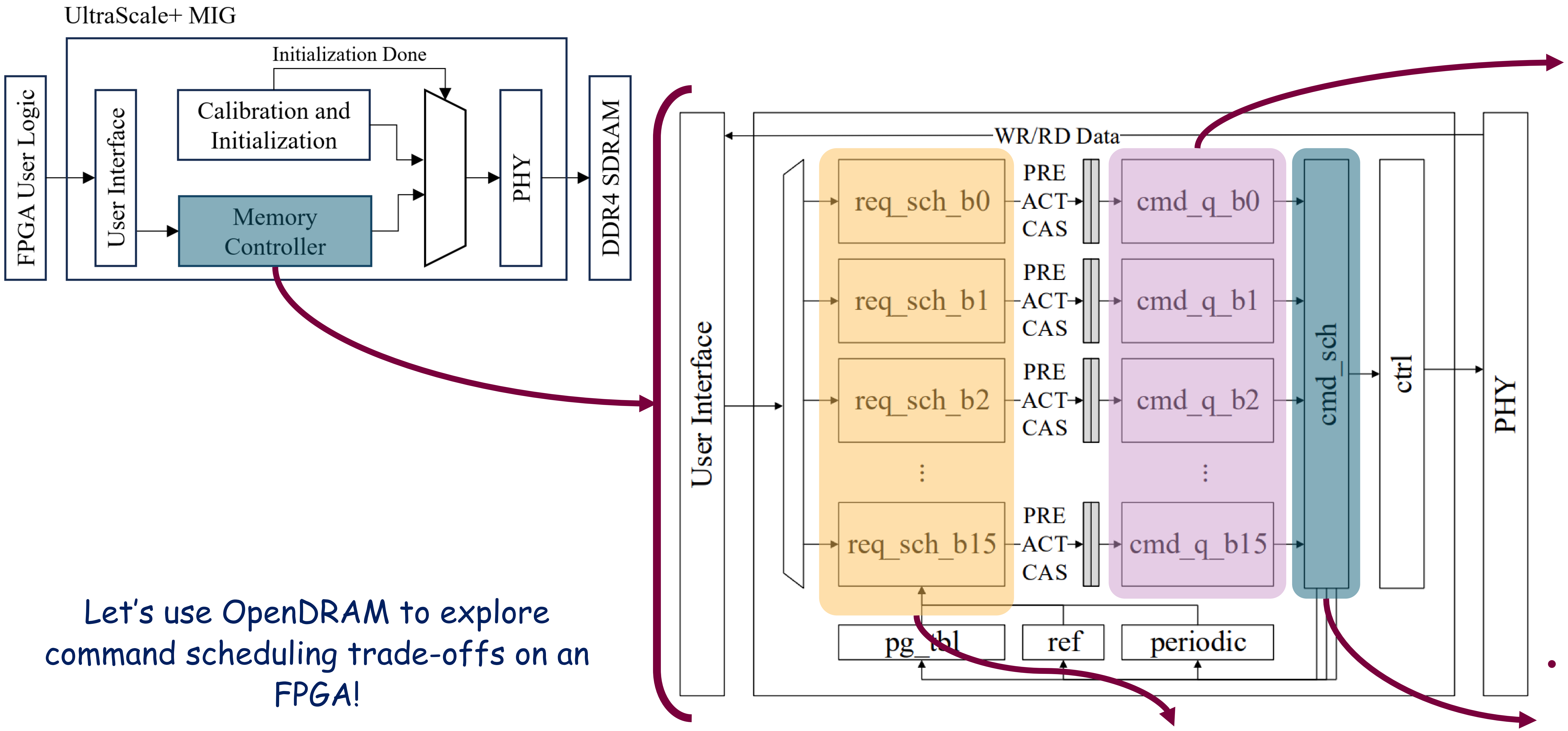


- Notice that four commands are sent to the PHY in each packet/cycle!
- Memory controller and PHY operate in different clock domains
  - MC is 4 times slower in our case
- If you are x4 slower, give me x4 more commands

OpenDRAM Architecture

Focused on Modularity and Extensibility

VIEW GITHUB REPOSITORY



- Per-bank Command Queues
- Prioritize open requests over closed ones

- Schedule 4 commands from Command Queues
- Inter-bank Round-Robin arbitration
- Track timing constraints

- Per-bank Request Queues
- FR-FCFS intra-bank arbiters

Let's use OpenDRAM to explore command scheduling trade-offs on an FPGA!

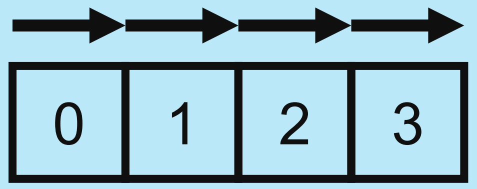
Use Case Demonstration

Command Scheduler Trade-offs

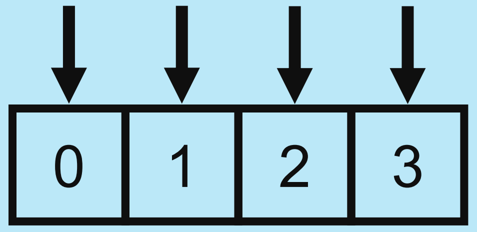
VIEW GITHUB REPOSITORY

How to pack 4 commands?

Consecutive Scheduling:



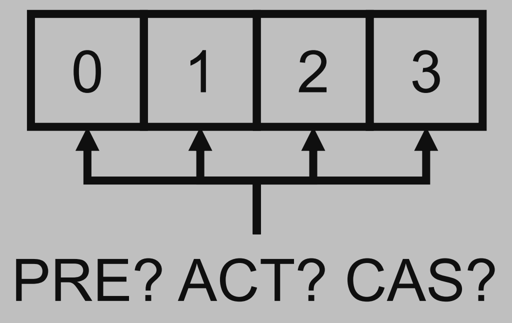
Parallel Scheduling:



Allocation Per Command Type:

PRE	PRE	CAS	ACT
CAS	ACT	CAS/ PRE	ACT/ PRE

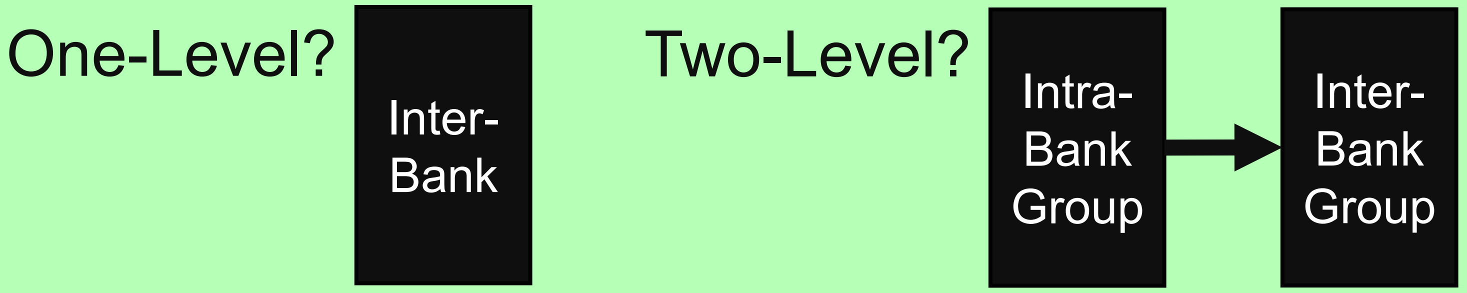
Challenge: Packing multiple commands.



Design Choice: A lot!

Trade-off: Performance, frequency, area, power, etc.

How to arbitrate among banks?

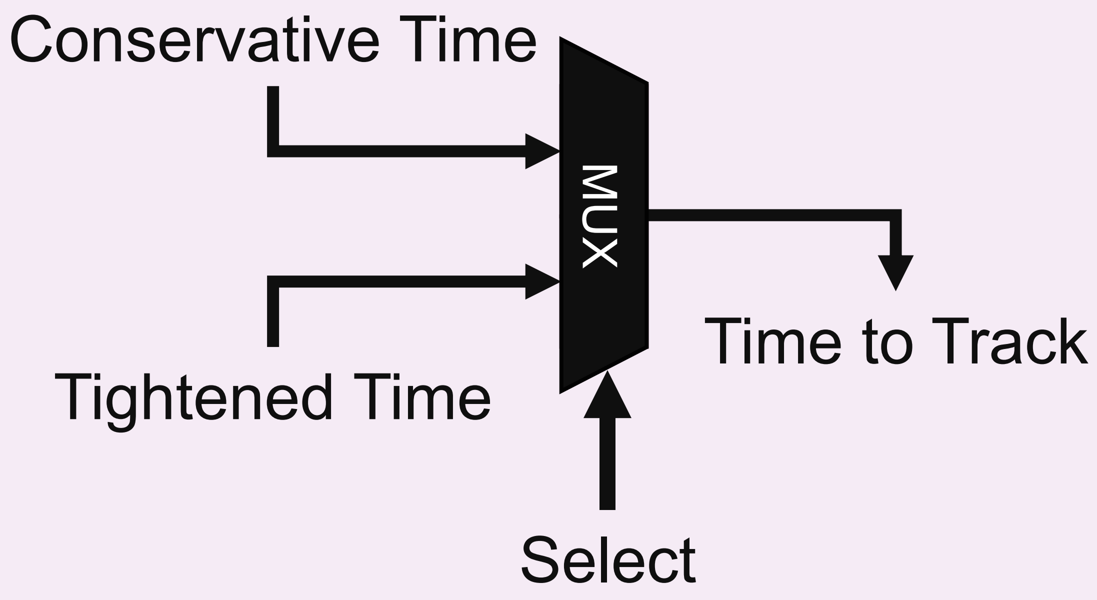


Design space is huge!

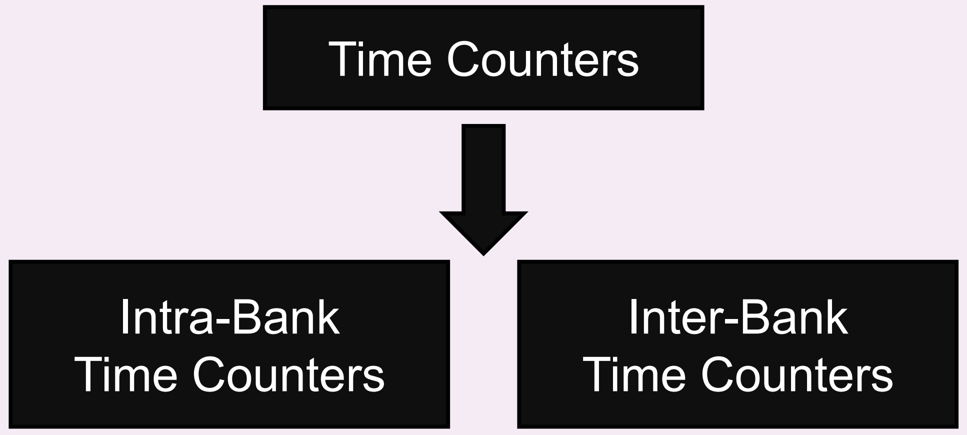
Impossible to evaluate without OpenDRAM!

How to track timing constraints?

Only Track the Conservative Ones:



Track in Separate Groups:



VIEW GITHUB REPOSITORY

Use Case Demonstration

Scheduling Trade-offs - Summary

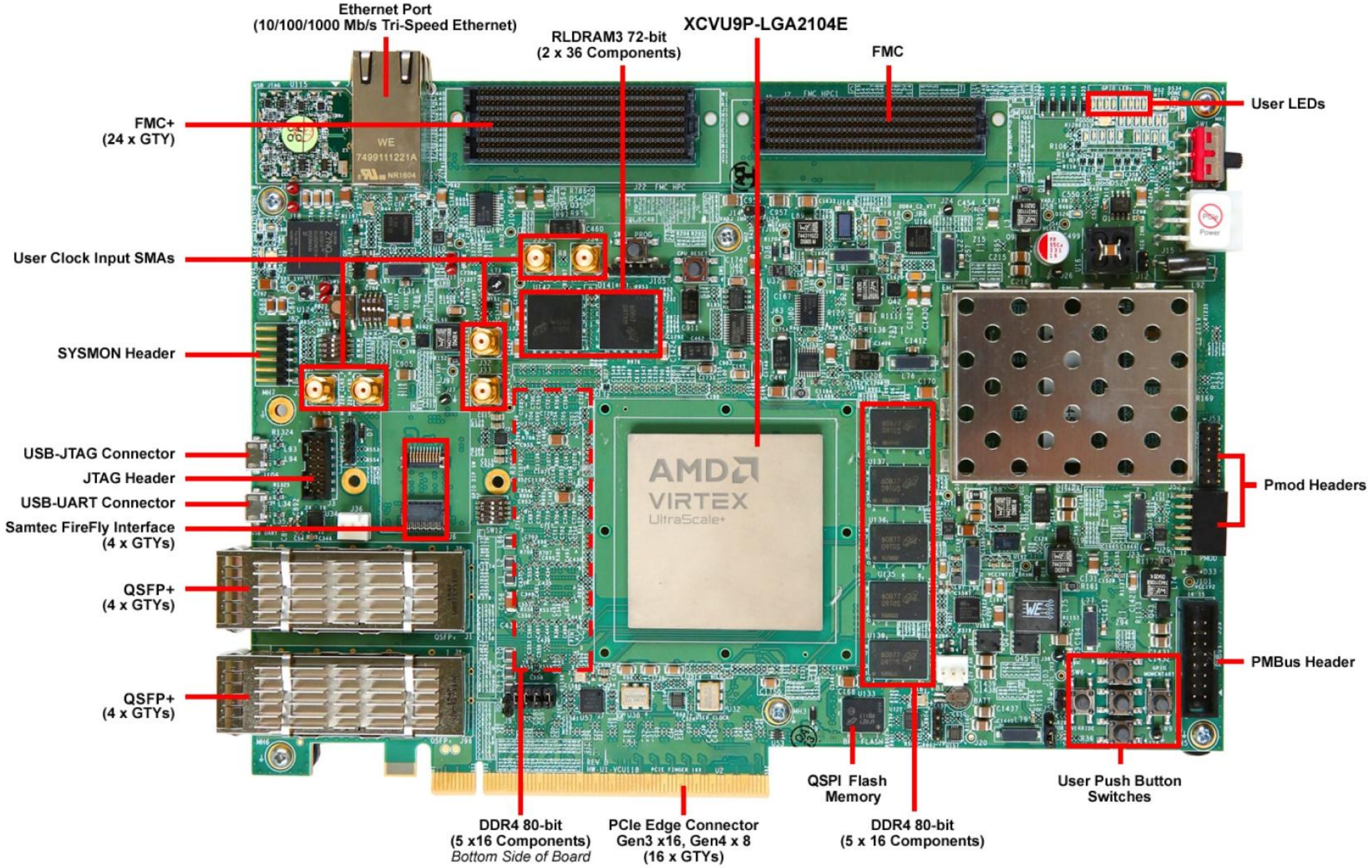
Frequency ← Performance →

Version	Unique Features	Traffic Affected
V1	• Sequential command arbitration and constraint tracker update • Issuing ACT and PRE in any slot	• n/a (no scheduling constraints.)
V2	• Separate arbiter per slot, operating in parallel • Constraint tracker updates once at the end • Issue at most two PRE • ACT issuance limited to slots 1 and 3	• May delay close requests.
V3	• Inflating t_{RRD_S} and t_{RRD_L} to be conservative and remove the t_{FAW} counter • At most one ACT and one CAS, but still two PRE	• May delay close requests.
V4	• Separate intra-bank and inter-bank tables, no need to compare when updating constraints within a bank • Increasing t_{RTP} and t_{WTR_S} to address corner cases	• May delay close requests after a read. • May delay consecutive read-write switching requests.
V5	• Two-stage command arbitration • Constraint tracker is in MC cycles, rather than DRAM cycles, reducing bits and comparison logic	• No added constraints.

Verification

VIEW GITHUB REPOSITORY

- Behavioural RTL simulation
- On-board verification using AMD Virtex UltraScale+ FPGA VCU118 Evaluation Kit
  - Using AMD MIG Advanced Traffic Generator (ATG)
    - Inject various traffic patterns and ensure data correctness
    - Validate initialization, calibration, and refresh
  - Full system verification using a RISC-V SoC



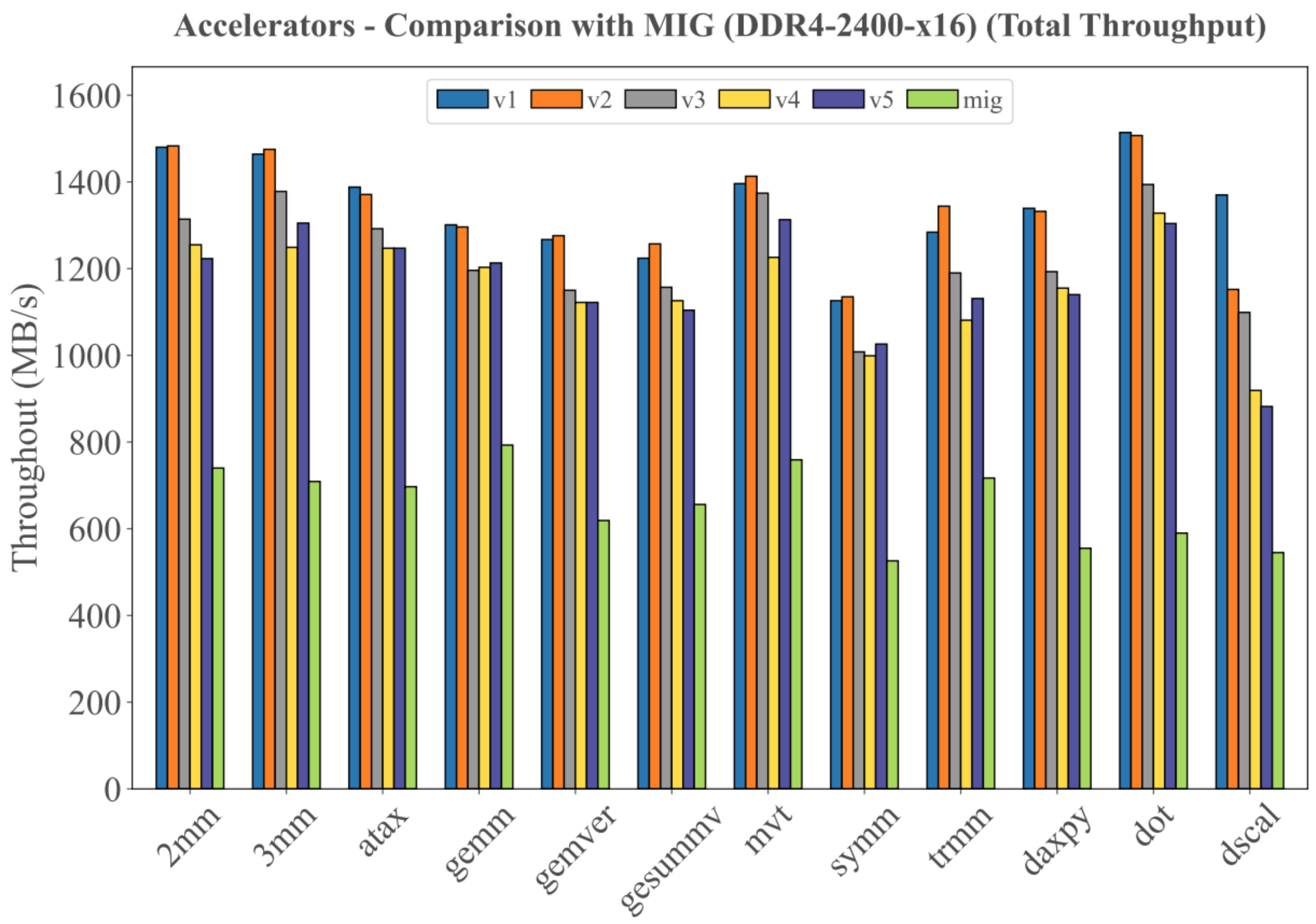
Picture Source:
https://www.amd.com/en/products/adaptive-socs-and-fpgas/evaluation-boards/vcu118.html#tabs-5d0c258eab-item-e37b42e889-tab

Evaluation

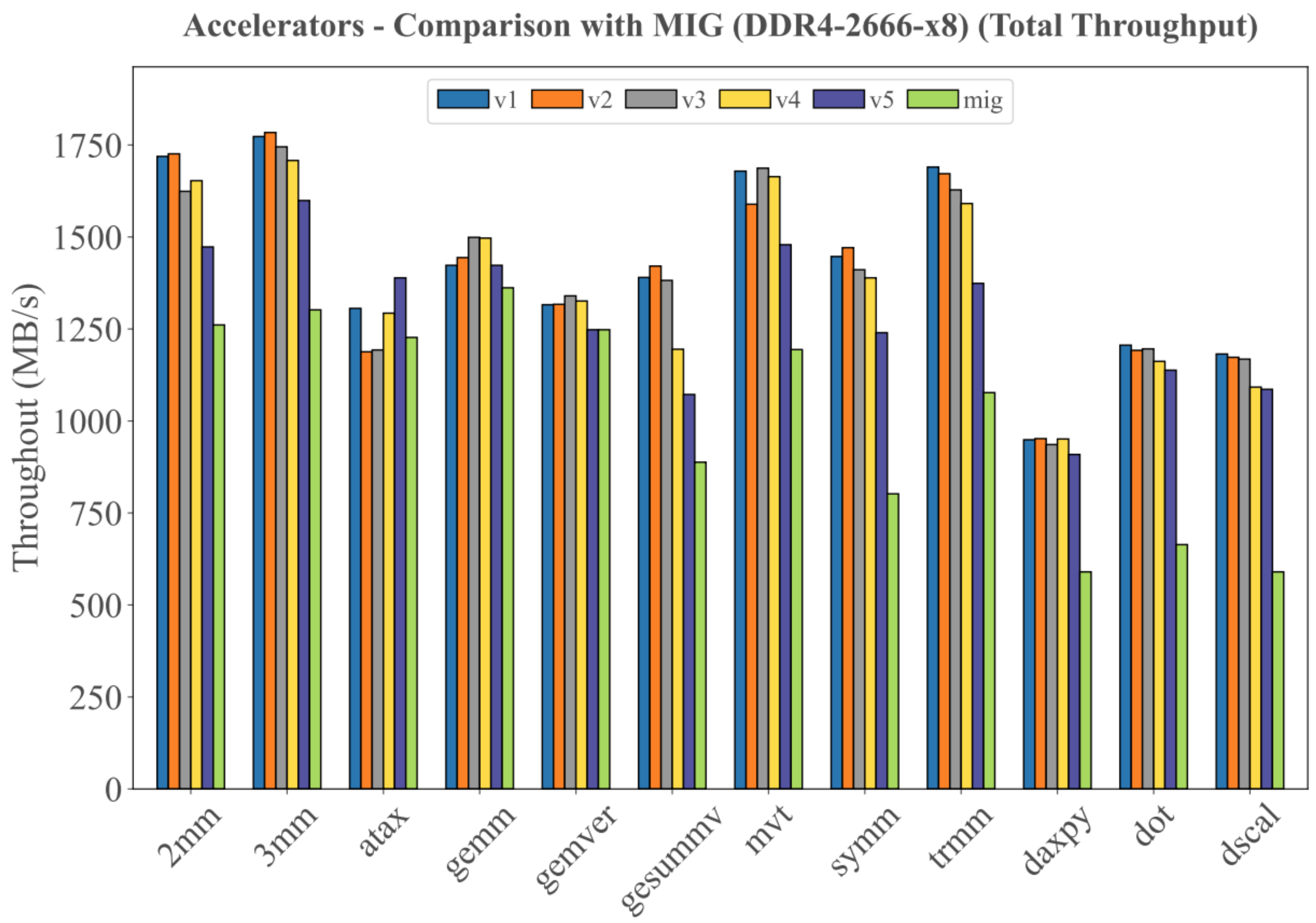
OpenDRAM vs. AMD MIG

VIEW GITHUB REPOSITORY

- Employed commonly deployed machine learning and scientific computing kernels.
- Emulated a four-accelerator system, all running the same kernel (different offset is added to the addresses from each kernel).
- With **8 banks**, providing approximately **77% to 107%** higher throughput across different versions (**Yes! Doubled!** 😱)
- With **16 banks**, achieving a throughput increase of between **35% to 47%**!



Accelerators - Comparison with MIG (DDR4-2400-x16) (Total Throughput)



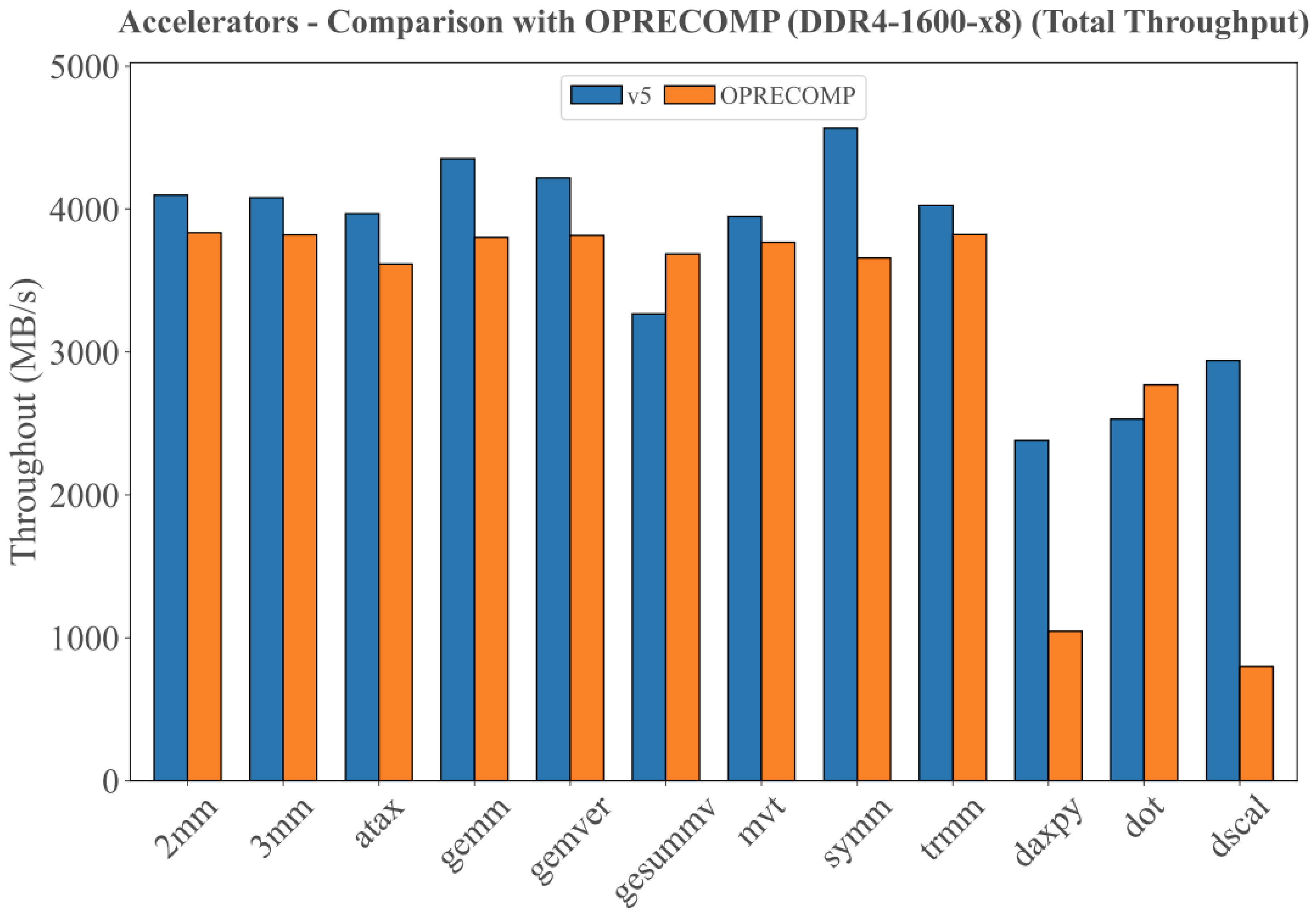
Accelerators - Comparison with MIG (DDR4-2666-x8) (Total Throughput)

Evaluation

OpenDRAM vs. OPRECOMP

VIEW GITHUB REPOSITORY

- The same setup, emulating four accelerators.
- Extensibility of our design allowed us to match the OPRECOMP DDR4 part
- With **16 banks**, an average performance improvement of **37%**

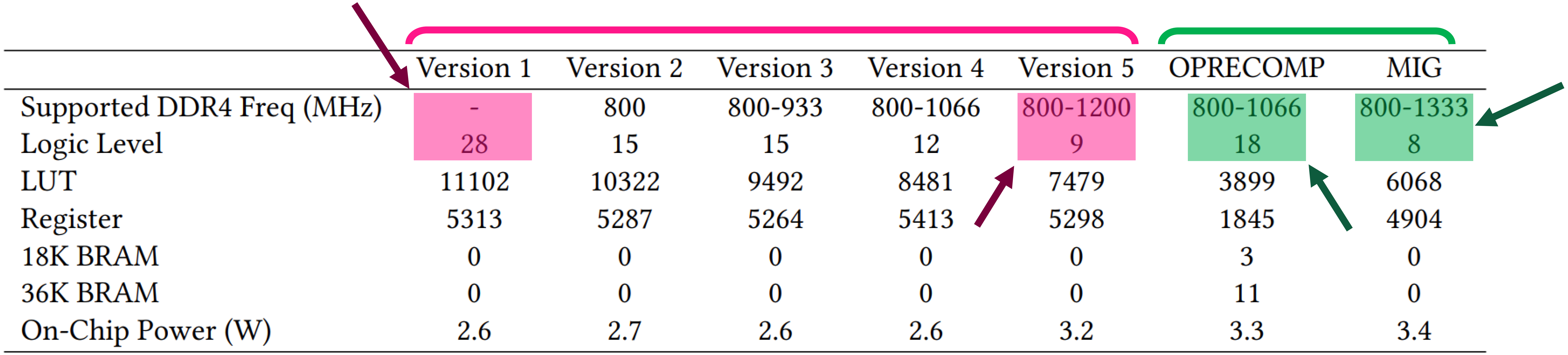


Evaluation

Implementation

VIEW GITHUB REPOSITORY

- Platform: AMD UltraScale+ VCU118 REV 2.0
- Device: 8-bank DDR4-2400-x16



	Version 1	Version 2	Version 3	Version 4	Version 5	OPRECOMP	MIG
Supported DDR4 Freq (MHz)	-	800	800-933	800-1066	800-1200	800-1066	800-1333
Logic Level	28	15	15	12	9	18	8
LUT	11102	10322	9492	8481	7479	3899	6068
Register	5313	5287	5264	5413	5298	1845	4904
18K BRAM	0	0	0	0	0	3	0
36K BRAM	0	0	0	0	0	11	0
On-Chip Power (W)	2.6	2.7	2.6	2.6	3.2	3.3	3.4

Conclusion and Future Work

VIEW GITHUB REPOSITORY

- OpenDRAM: the first modular, open-source, soft FPGA memory controller for DDR4 SDRAM
- Performance gains of up to **157% over AMD MIG** and up to **267% over OPRECOMP**
- Thoroughly verified on hardware
- As a use case, five different versions of the Command Scheduler were developed to study the trade-offs
- In future, we plan to investigate:
  - Mapping to DDR5 technology
    - We did not have access to an FPGA equipped with a DDR5 PHY capable of bypassing the vendor-provided memory controller.
  - Mapping to Intel FPGAs
  - Integrating performance counters

VIEW GITHUB REPOSITORY

Thank you for your kind attention. 😳

We'll be answering questions.

Email: abbasa46@mcmaster.ca

UNIVERSITY OF WATERLOO

McMaster University