MICRO-ARCHITECTURAL EXPLORATION OF THE RELATIONAL MEMORY ENGINE (RME) ON RISC-V SOC USING FIRESIM

Cole Strickler, Ju Hyoung Mun, Connor Sullivan, Denis Hoornaert, Renato Mancuso, Manos Athanassoulis, Heechul Yun

University of Kansas, Brandeis University, Boston University, TU Munich

KU-CSL
COMPUTER SYSTEMS LAB

Sixteenth International Workshop on Accelerating Analytics and Data Management Systems Using Modern Processor and Storage Architectures

MOTIVATION

- Data movement is the bottleneck[1]
- Optimal layout reduces data movement
  - Row store
  - Column Store
  - Hybrid-Transactional Analytical Processing (HTAP)

[1] Manos Frouzakis, Juan Gómez-Luna, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu. 2025. PIMDAL: Mitigating the Memory Bottleneck in Data Analytics using a Real Processing-in-Memory System. CoRR abs/2504.01948 (2025). https://doi.org/10.48550/arXiv.2504.01948

MOTIVATION

- Data movement is the bottleneck[1]
- Optimal layout reduces data movement
  - Row store
  - Column Store
  - Hybrid-Transactional Analytical Processing (HTAP)

HTAP Issue #1

- Data Duplication

[1] Manos Frouzakis, Juan Gómez-Luna, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu. 2025. PIMDAL: Mitigating the Memory Bottleneck in Data Analytics using a Real Processing-in-Memory System. CoRR abs/2504.01948 (2025). https://doi.org/10.48550/arXiv.2504.01948

MOTIVATION

- Data movement is the bottleneck[1]
- Optimal layout reduces data movement
  - Row store
  - Column Store
  - Hybrid-Transactional Analytical Processing (HTAP)

HTAP Issue #1

- Data Duplication

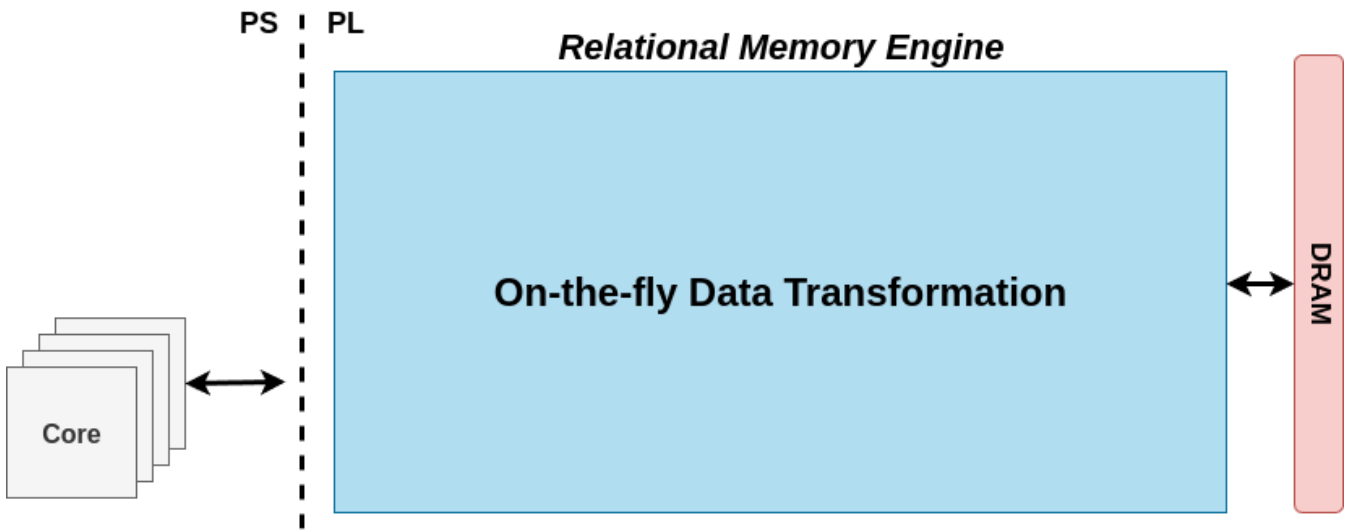
HTAP Issue #2

- Code complexity

[1] Manos Frouzakis, Juan Gómez-Luna, Geraldo F. Oliveira, Mohammad Sadrosadati, and Onur Mutlu. 2025. PIMDAL: Mitigating the Memory Bottleneck in Data Analytics using a Real Processing-in-Memory System. CoRR abs/2504.01948 (2025). https://doi.org/10.48550/arXiv.2504.01948

RELATIONAL MEMORY ENGINE (RME)[1]

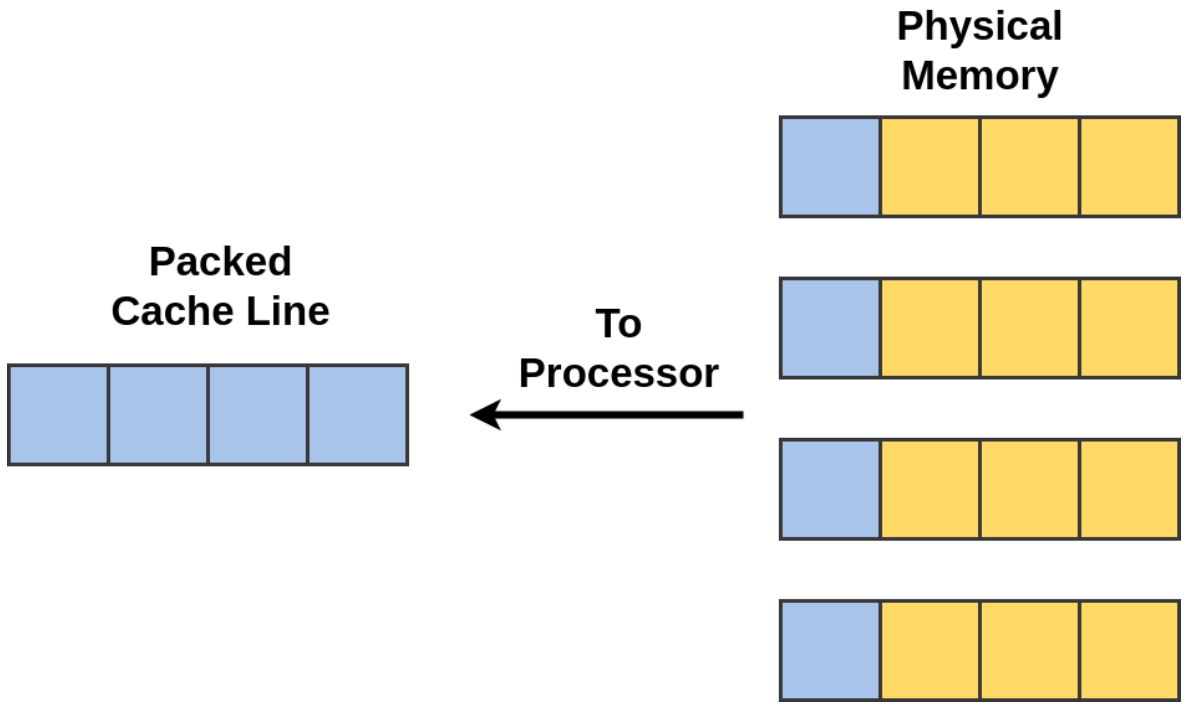
- Implements on-the-fly data transformation
- Benefits of HTAP but with single data layout



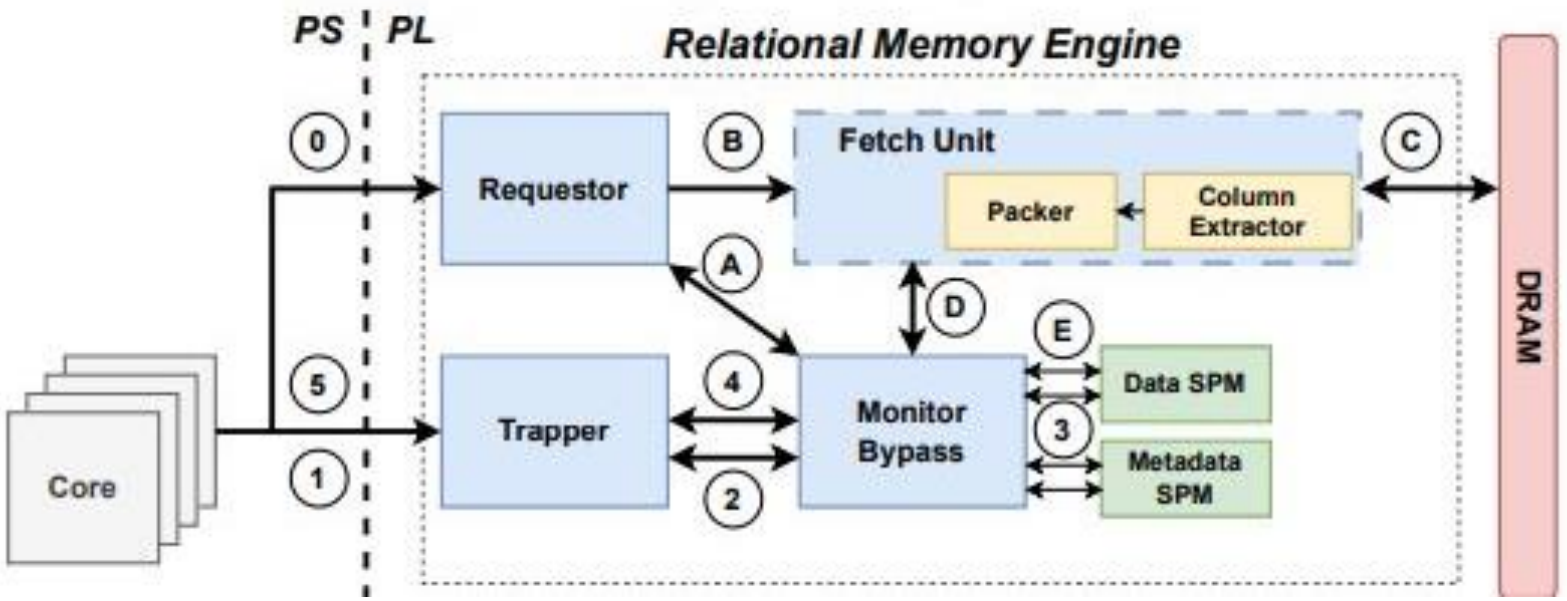
[1] Shahin Roozkhosh, Denis Hoornaert, Ju Hyoung Mun, Tarikul Islam Papon, Ahmed Sanaullah, Ulrich Drepper, Renato Mancuso, and Manos Athanassoulis. 2023. Relational Memory: Native In-Memory Accesses on Rows and Columns. In Proceedings of the International Conference on Extending Database Technology (EDBT). 66–79. https://doi.org/10.48786/edbt.2023.06

ON-THE-FLY DATA TRANSFORMATION

- Always present optimal layout to CPU
- Avoid useless data movement
- Achieve ideal locality

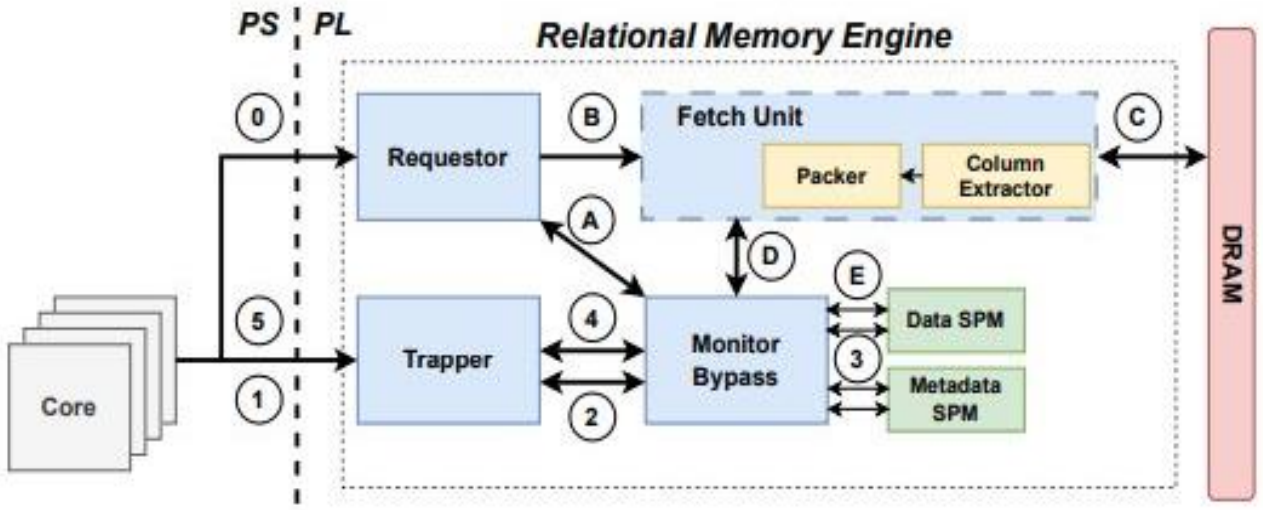


RELATIONAL MEMORY ENGINE



RELATIONAL MEMORY ENGINE

- Limitations:
  - PS-PL FPGA Platform[1]
  - Fixed 100MHz frequency
  - FPGA specific resources (i.e. BRAM)
  - Fixed hardware platform



[1] Advanced Micro Devices. 2025. AMD Zynq™ UltraScale+™ MPSoCs. Technical Report. Advanced Micro Devices. https://www.amd.com/en/products/adaptivesocs-and-fpgas/soc/zynq-ultrascale-plus-mpsoc.html

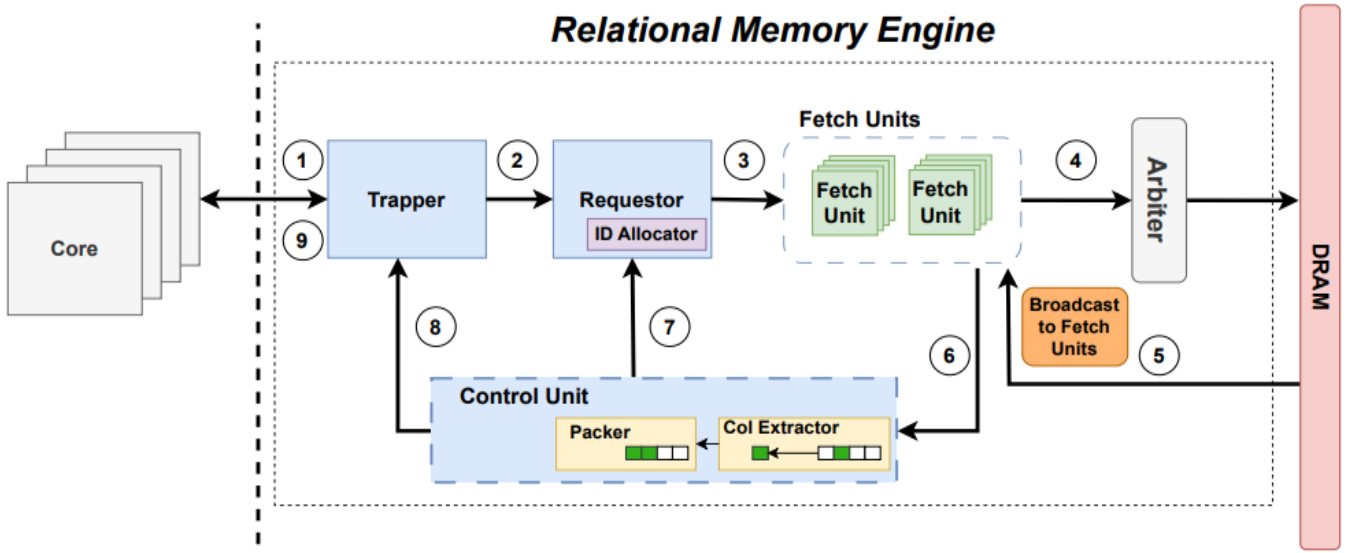
OUR WORK

- Re-implement and improve RME in Chisel[1] HDL
- Integrate into RISC-V SoC and FireSim[2] infrastructure
- Perform micro-architectural exploration
- Validate data transformation paradigm

[1] Jonathan Bachrach, Huy Vo, Brian C. Richards, Yunsup Lee, Andrew Waterman, Rimas Avizienis, John Wawrzynek, and Krste Asanovic. 2012. Chisel: Constructing Hardware in a Scala Embedded Language. In Proceedings of the Annual Design Automation Conference (DAC). 1216–1225. https://doi.org/10. 1145/2228360.2228584
[2] Sagar Karandikar, Howard Mao, Donggyu Kim, David Biancolin, Alon Amid, Dayeol Lee, Nathan Pemberton, Emmanuel Amaro, Colin Schmidt, Aditya Chopra, Qijing Huang, Kyle Kovacs, Borivoje Nikolic, Randy H. Katz, Jonathan Bachrach, and Krste Asanovic. 2018. FireSim: FPGA-Accelerated Cycle-Exact Scale-Out System Simulation in the Public Cloud. In Proceedings of the ACM/IEEE Annual International Symposium on Computer Architecture (ISCA). 29–42. https: //doi.org/10.1109/ISCA.2018.00014

OUR DESIGN

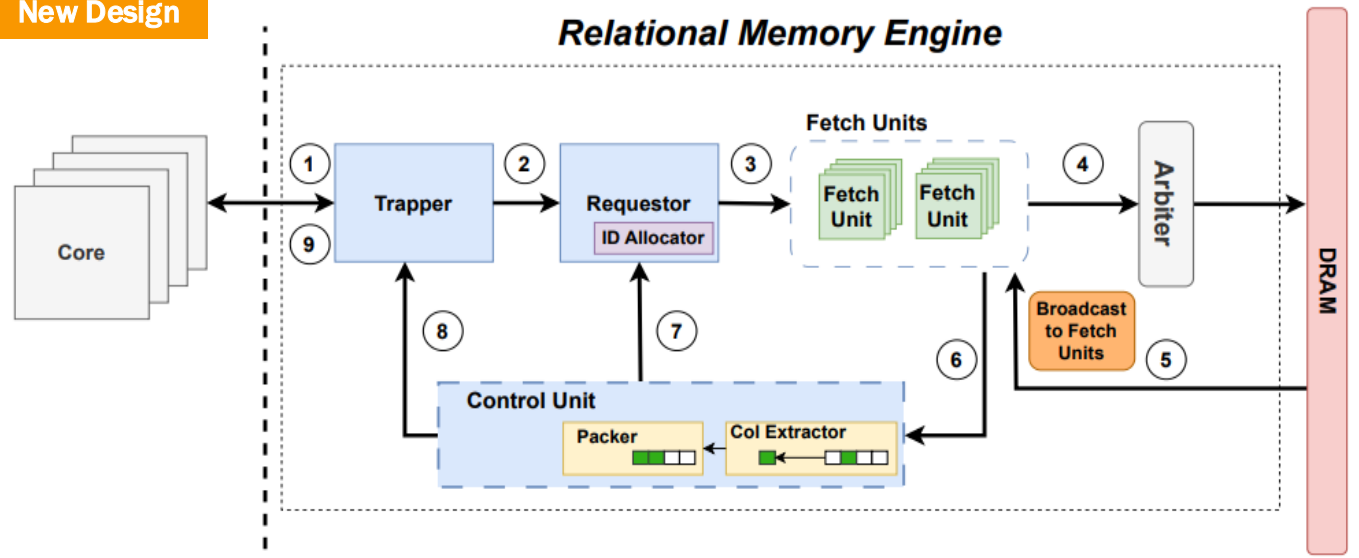
- Port from AXI -> TileLink
- Architectural improvements



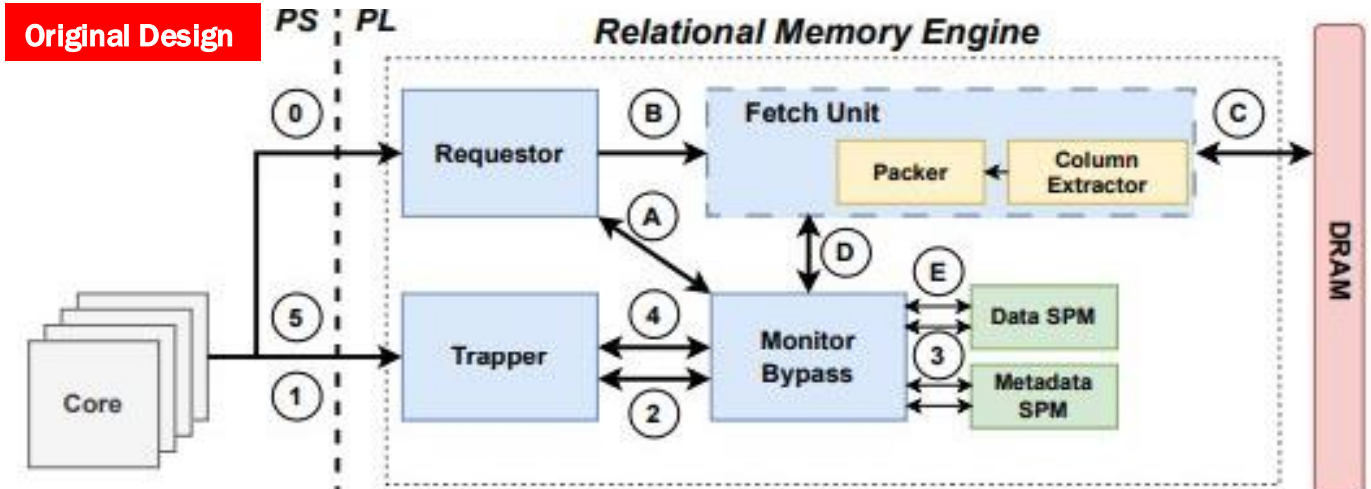
OUR DESIGN

- Improvements:
  - Removal of large SPMs
  - Out-of-order transaction handling
  - Simplified control flow

New Design



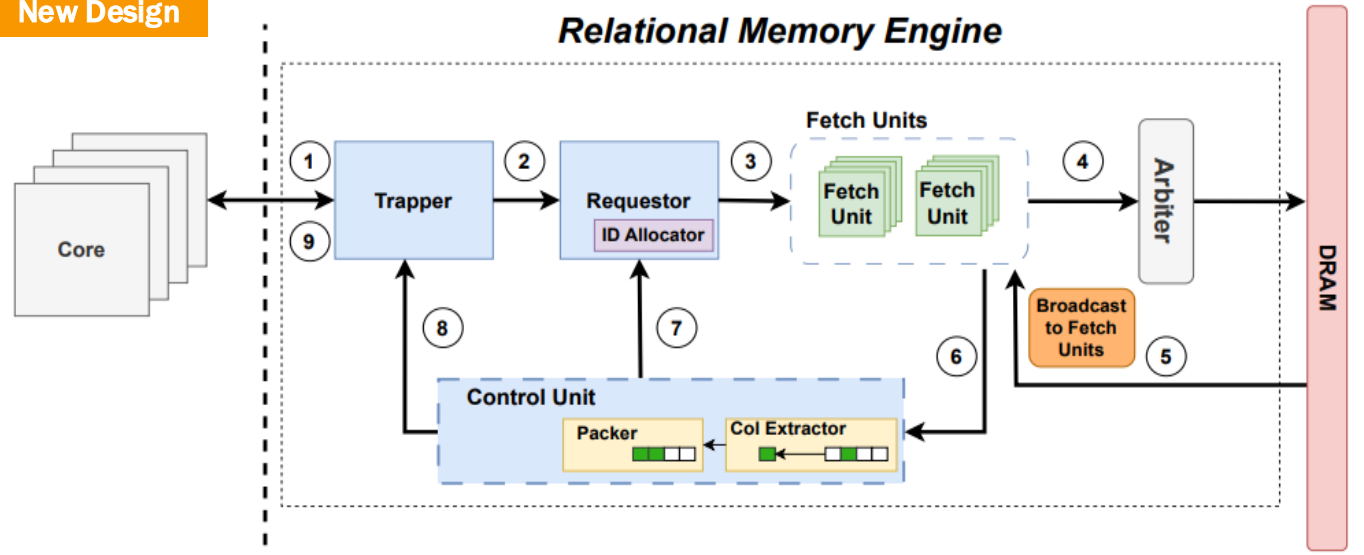
Original Design



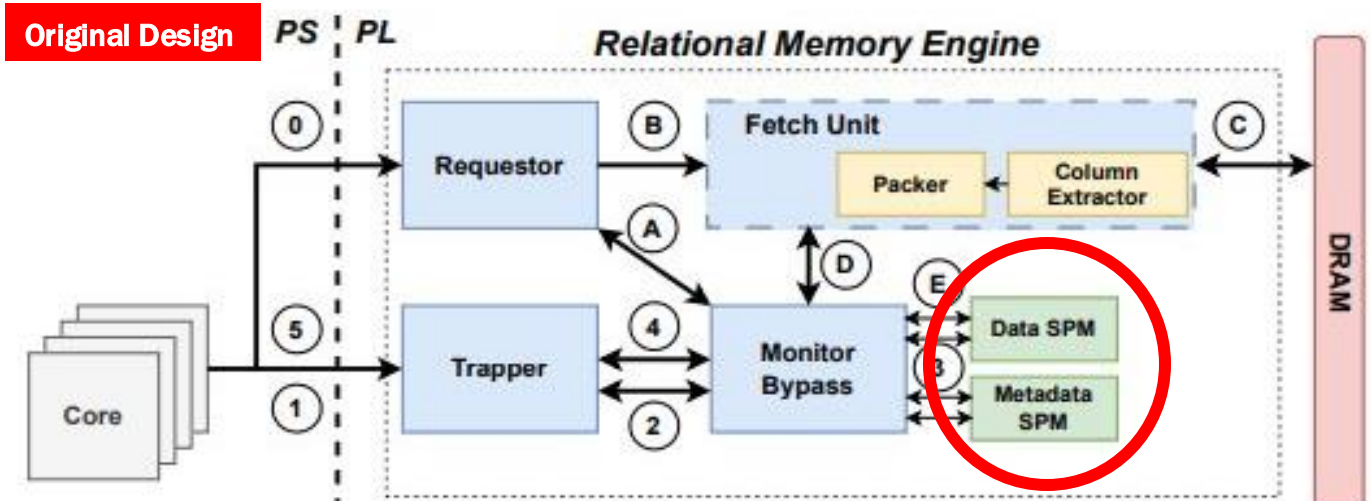
OUR DESIGN

- Improvements:
  - Removal of large SPMs
  - Out-of-order transaction handling
  - Simplified control flow

New Design



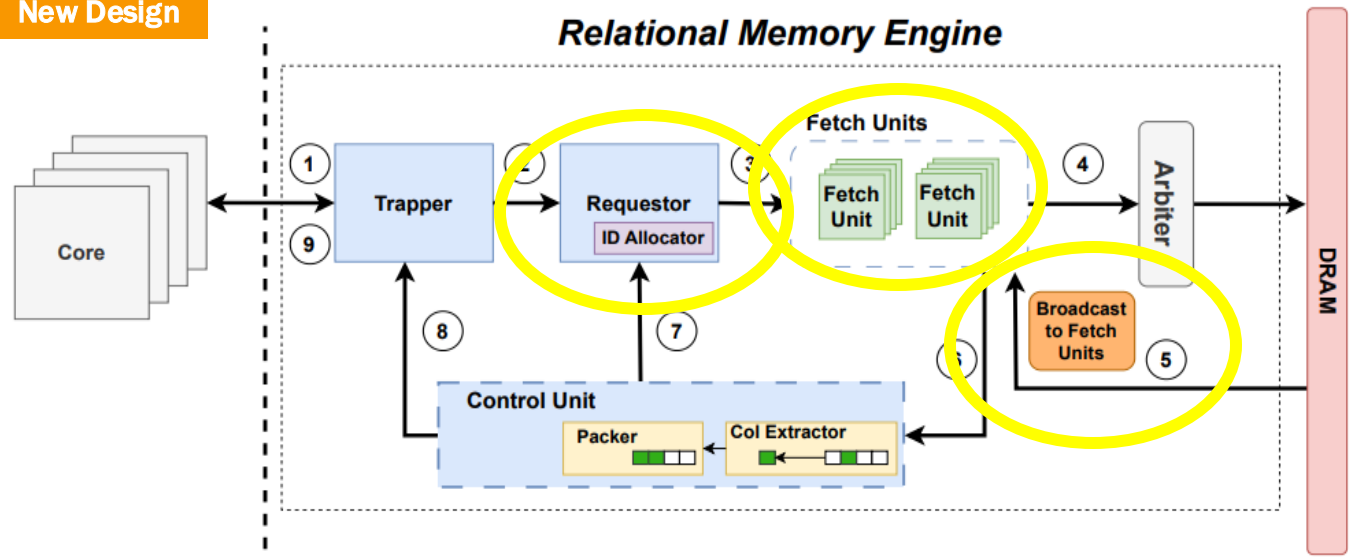
Original Design



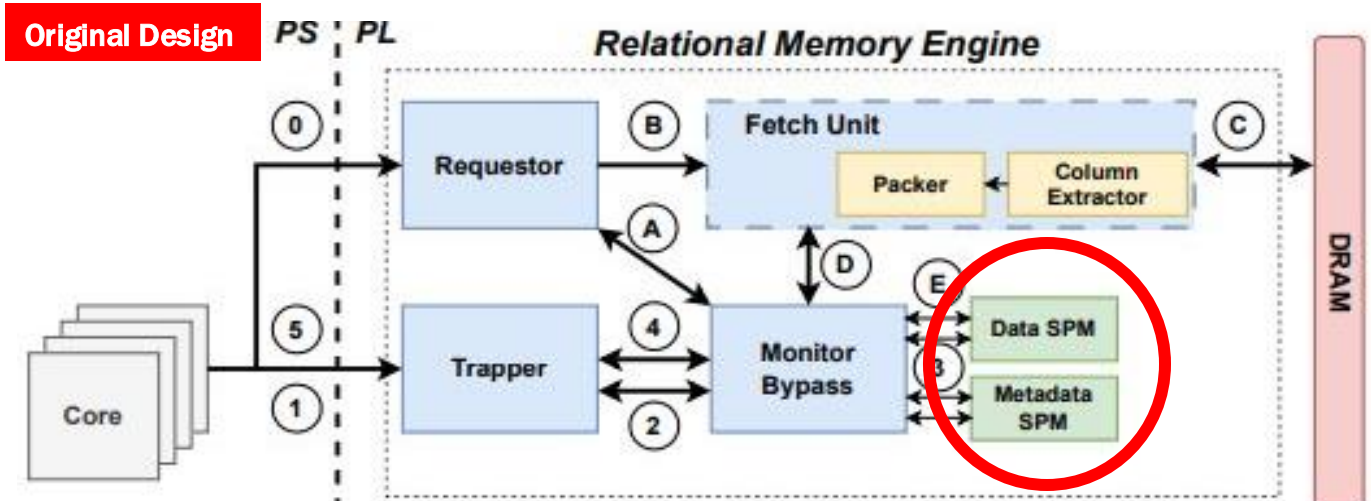
OUR DESIGN

- Improvements:
  - Removal of large SPMs
  - Out-of-order transaction handling
  - Simplified control flow

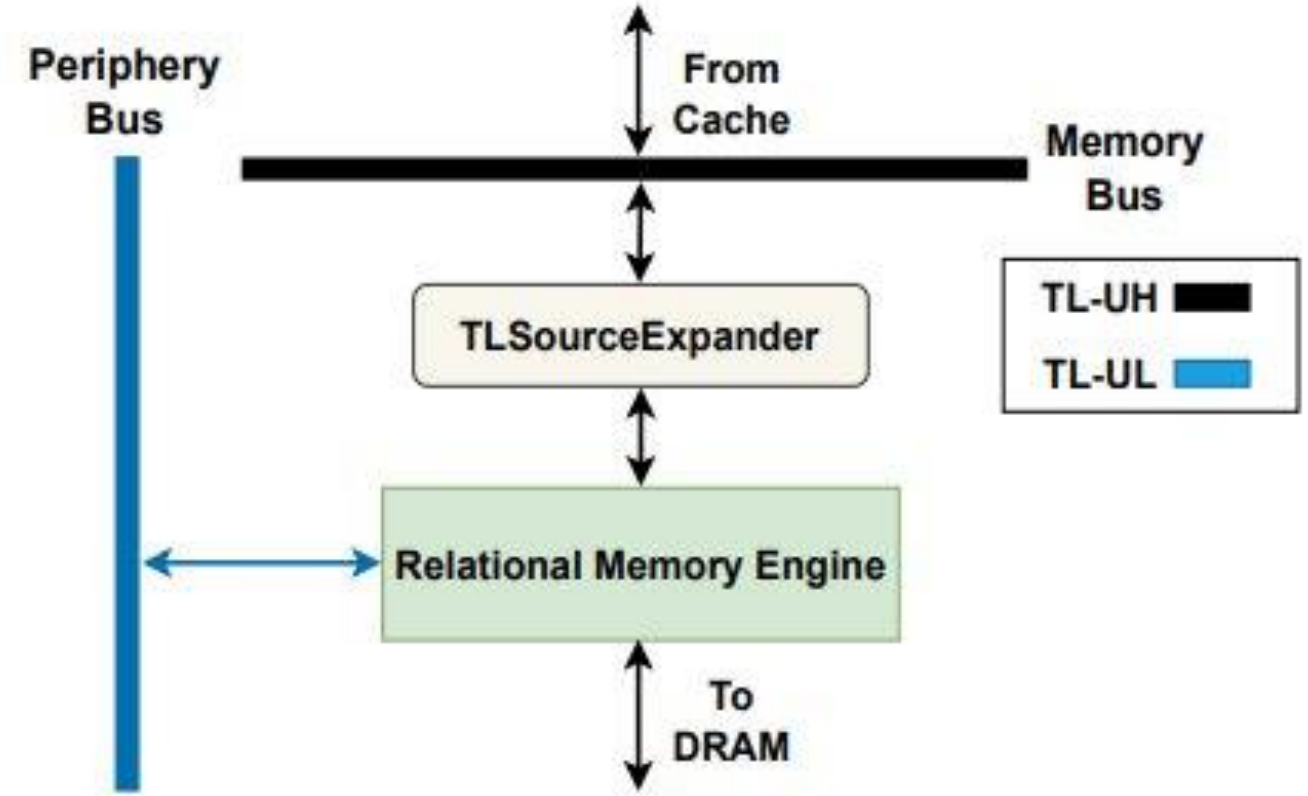
New Design



Original Design



OUR DESIGN


Periphery Bus
From Cache
Memory Bus
TL-UH
TL-UL
TLSourceExpander
Relational Memory Engine
To DRAM

MICRO-ARCHITECTURAL EXPLORATION

- What is impact of pre-fetching?
- What is impact of out-of-order core?
- What is impact of RME clock speed?

Table 1: Expanding RME evaluation setup.

Setup	Details
RME [23]	In-order core 1.5 GHz w/ prefetcher, RME @ 100MHz
Design 1	In-order core 1.0 GHz (w/ prefetcher, w/o prefetcher), RME @ 1 GHz
Design 2	Out-of-order core 1.0 GHz (w/ prefetcher, w/o prefetcher), RME @ 1 GHz
Design 3	In-order core 1.0 GHz w/ prefetcher, RME @ (1000, 500, 250, 167, 142, 125, 111, 100) MHz

PLATFORM

Table 2: Evaluation platform specifications

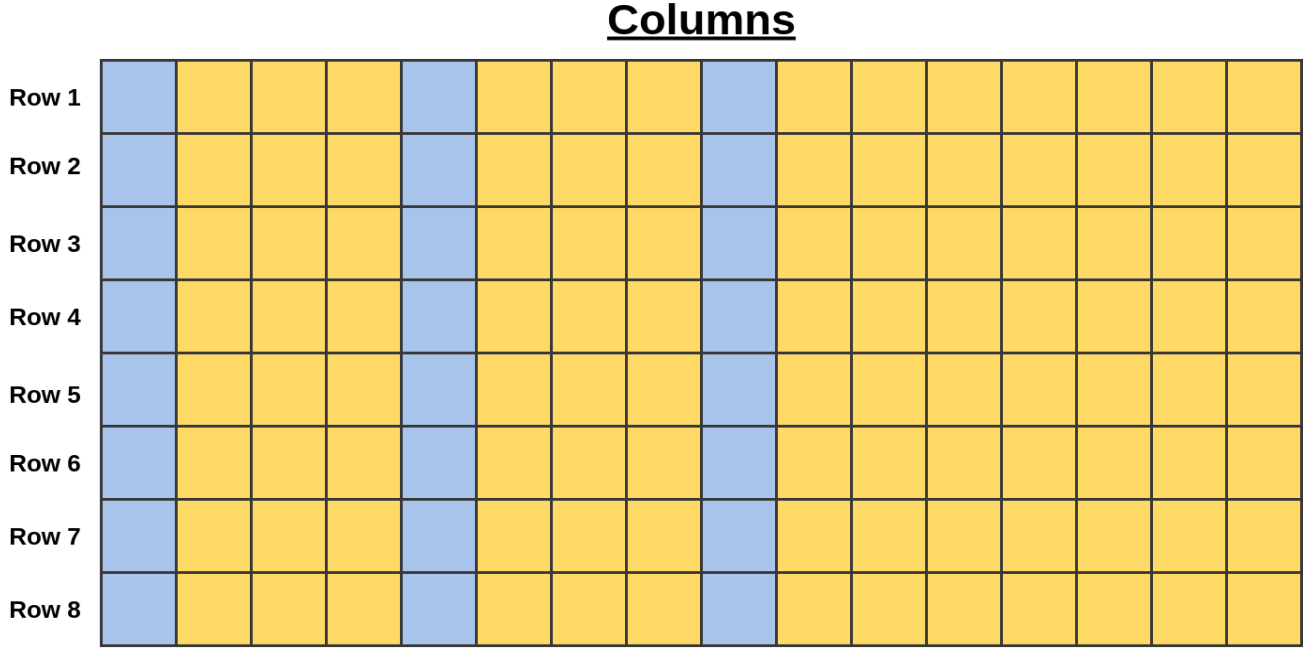
Component	Specifications
In-Order Core	Rocket, 1GHz, L1: 16K(I)/16K(D) (4-Way)
Out-of-Order Core	BOOM, 1GHz, 2-wide, ROB: 64, LSQ: 16/16, L1: 16K(I)/16K(D) (4-Way), 6 MSHRs
Shared L2 Cache	1MB (16-way), 1 Bank
Main Memory	4GB DDR3 1066 MHz, 1 Rank, 8 Banks
Prefetcher	L1 4-Ahead Multi Next Line
RME Parameters	16 Fetch Units
Operating System	Buildroot Linux v6.2.0

WORKLOAD

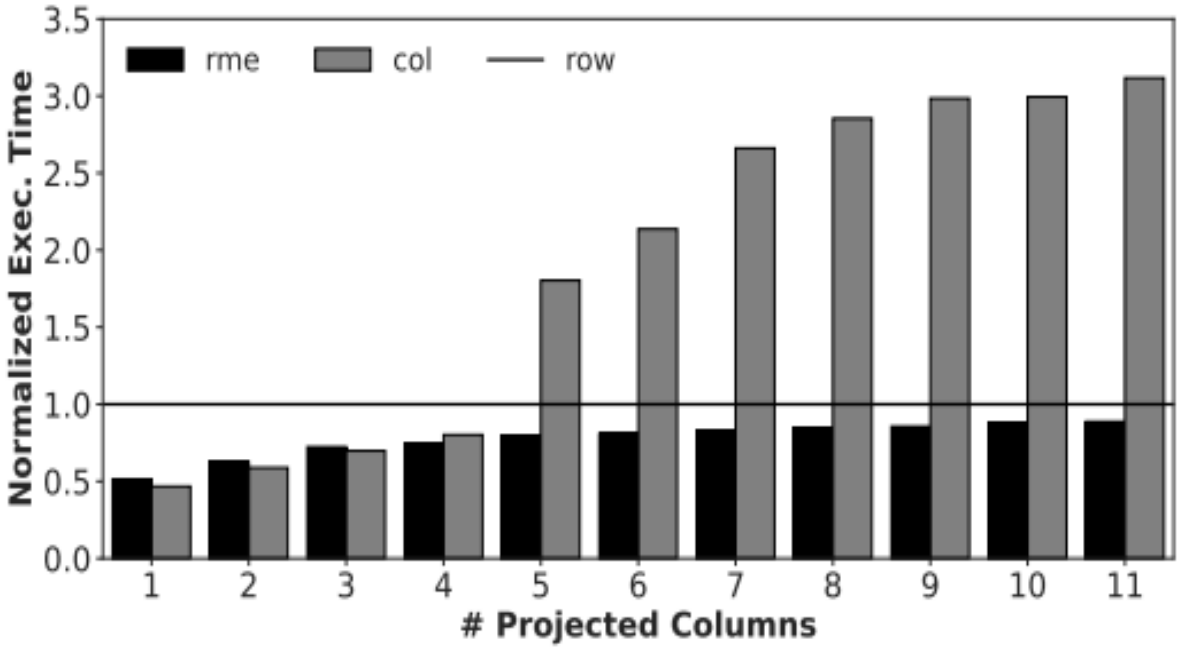
- In-memory database
- Fix row size to 64 bytes
- Q0 and Q1 maximally stress the RME, best exhibit data transformation

Listing 1: Evaluated Database Queries

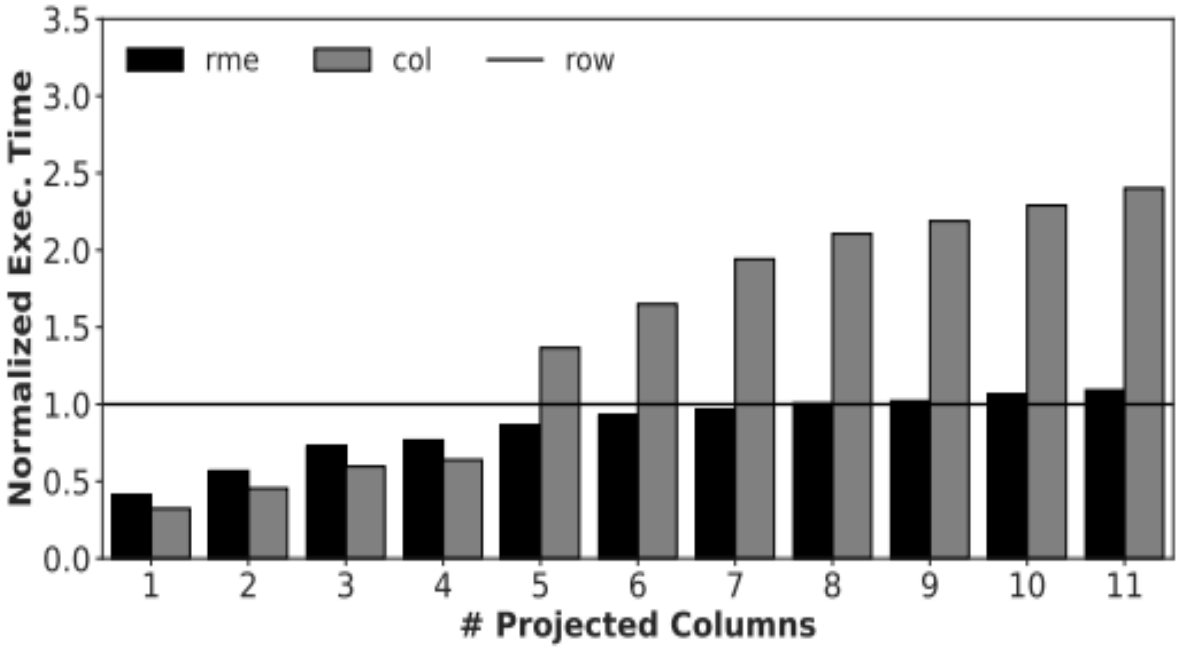
```
Q0: SELECT A1, A2, A3 FROM S;
Q1: SELECT A1, A2, ..., Ak FROM S;
```



EXP 1: EFFECT OF PREFETCHING



With Prefetcher

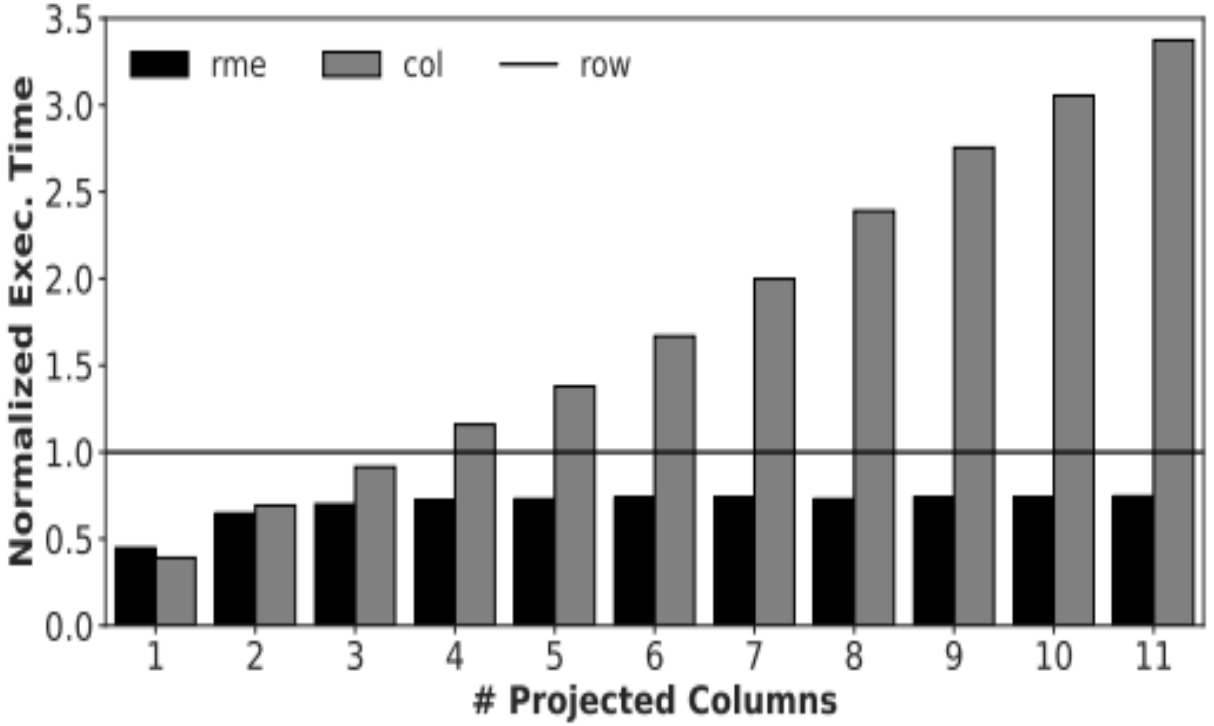


Without Prefetcher

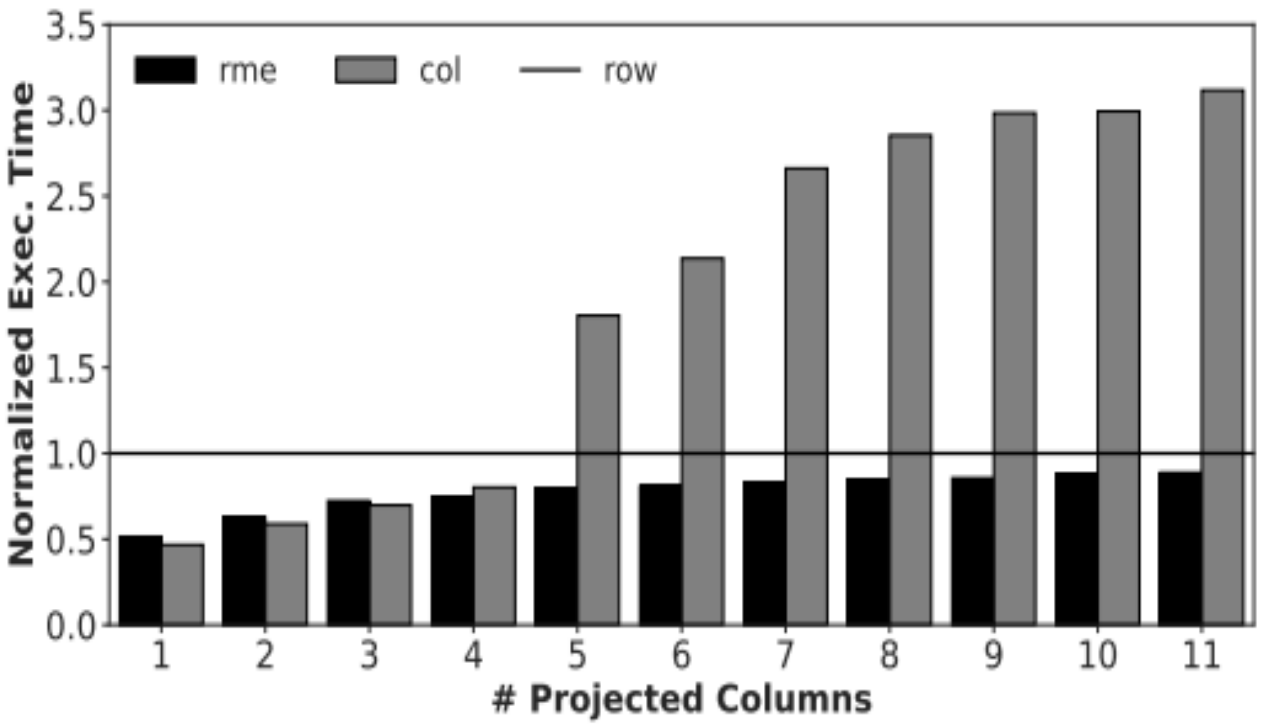
EXP 1: TAKEAWAYS

- Trends confirm results seen in original RME paper
  - Outperform both row & column store
- Prefetchers hide latency of RME

EXP 2: EFFECT OF OUT-OF-ORDER CORE

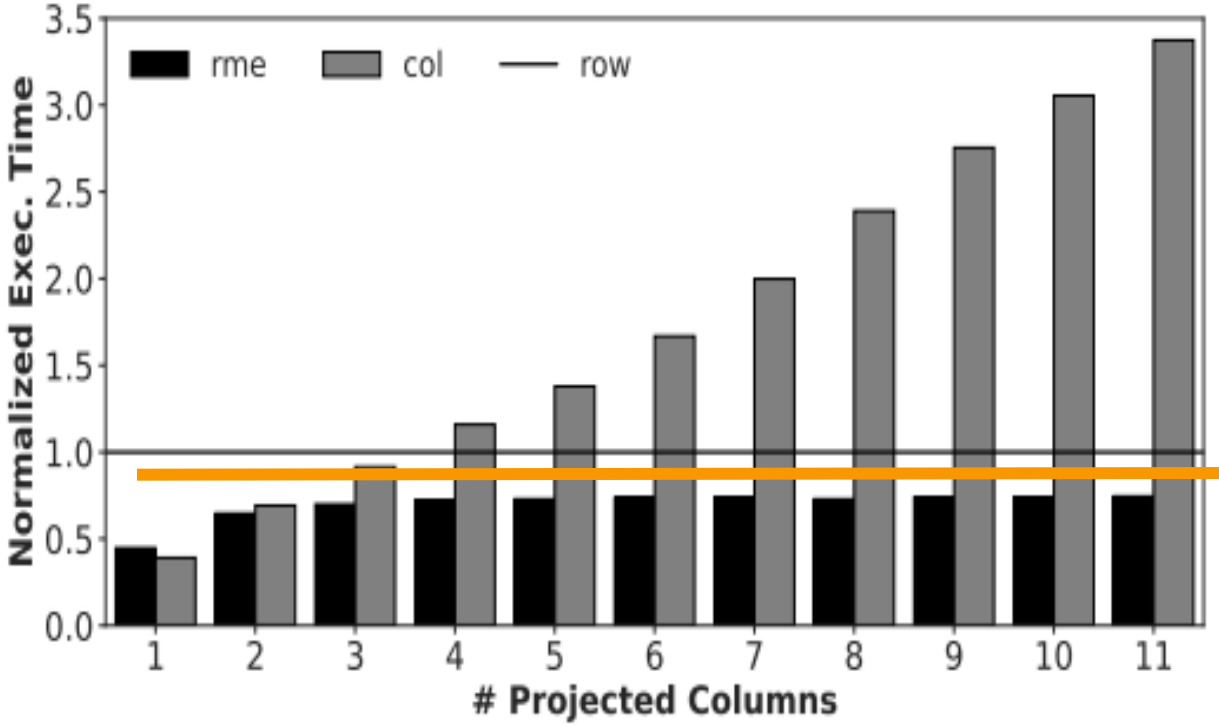


Out-of-order

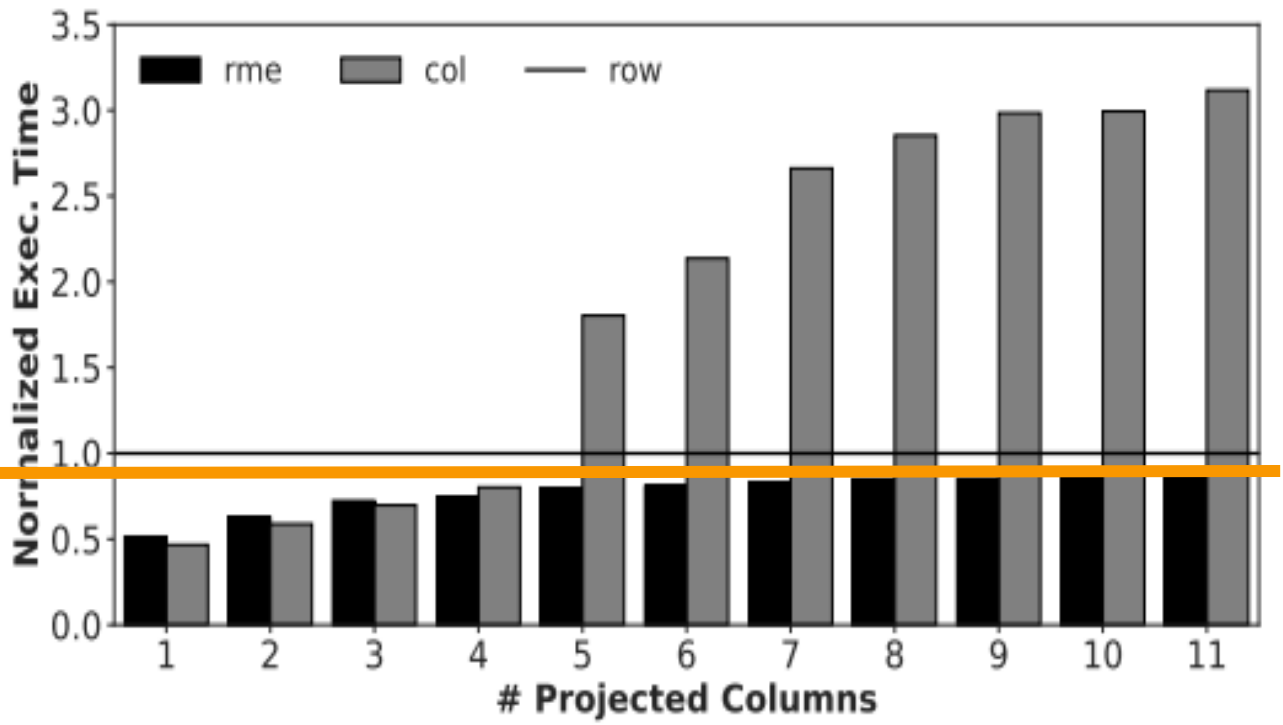


In-order

EXP 2: EFFECT OF OUT-OF-ORDER CORE



Out-of-order

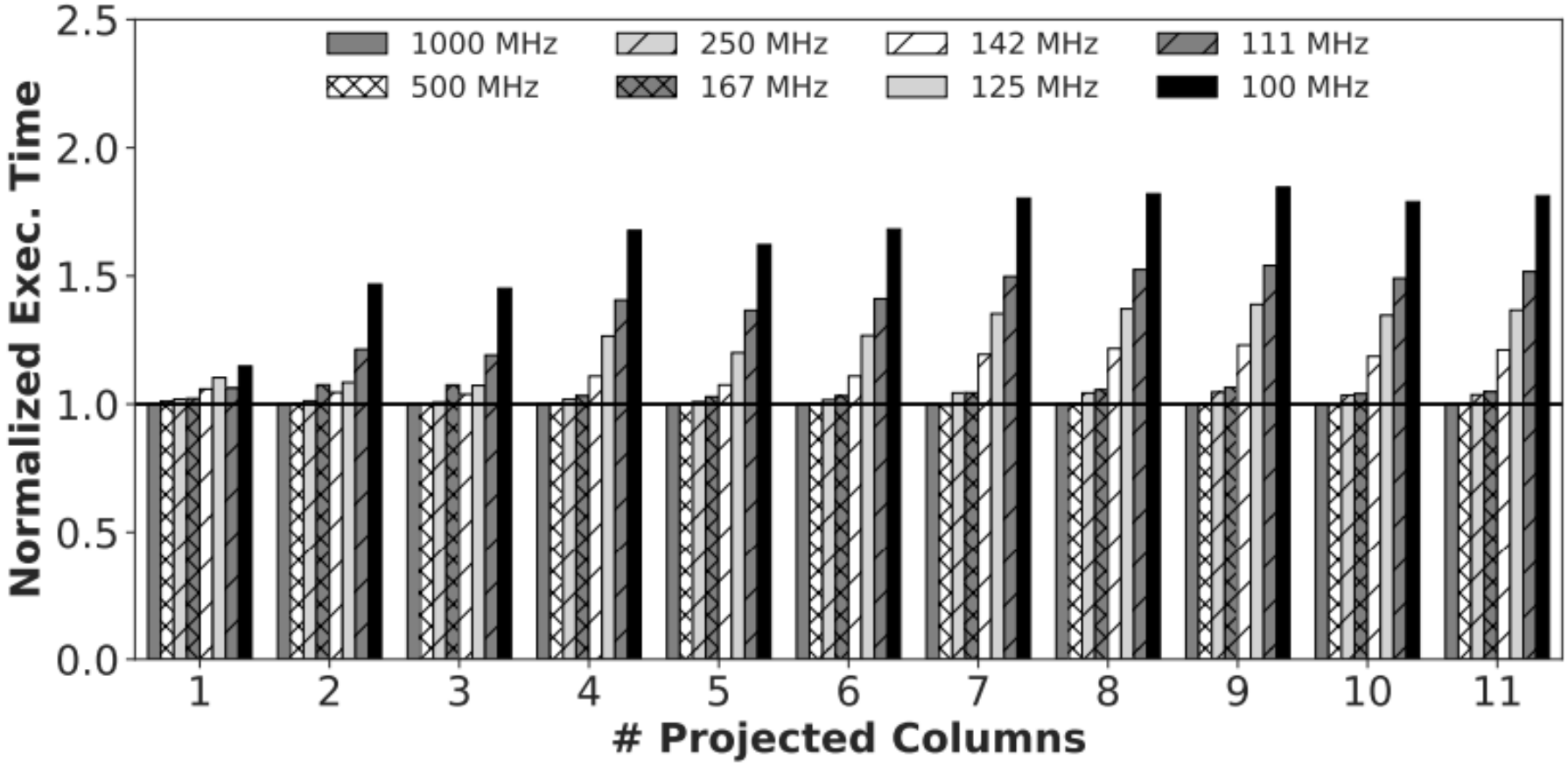


In-order

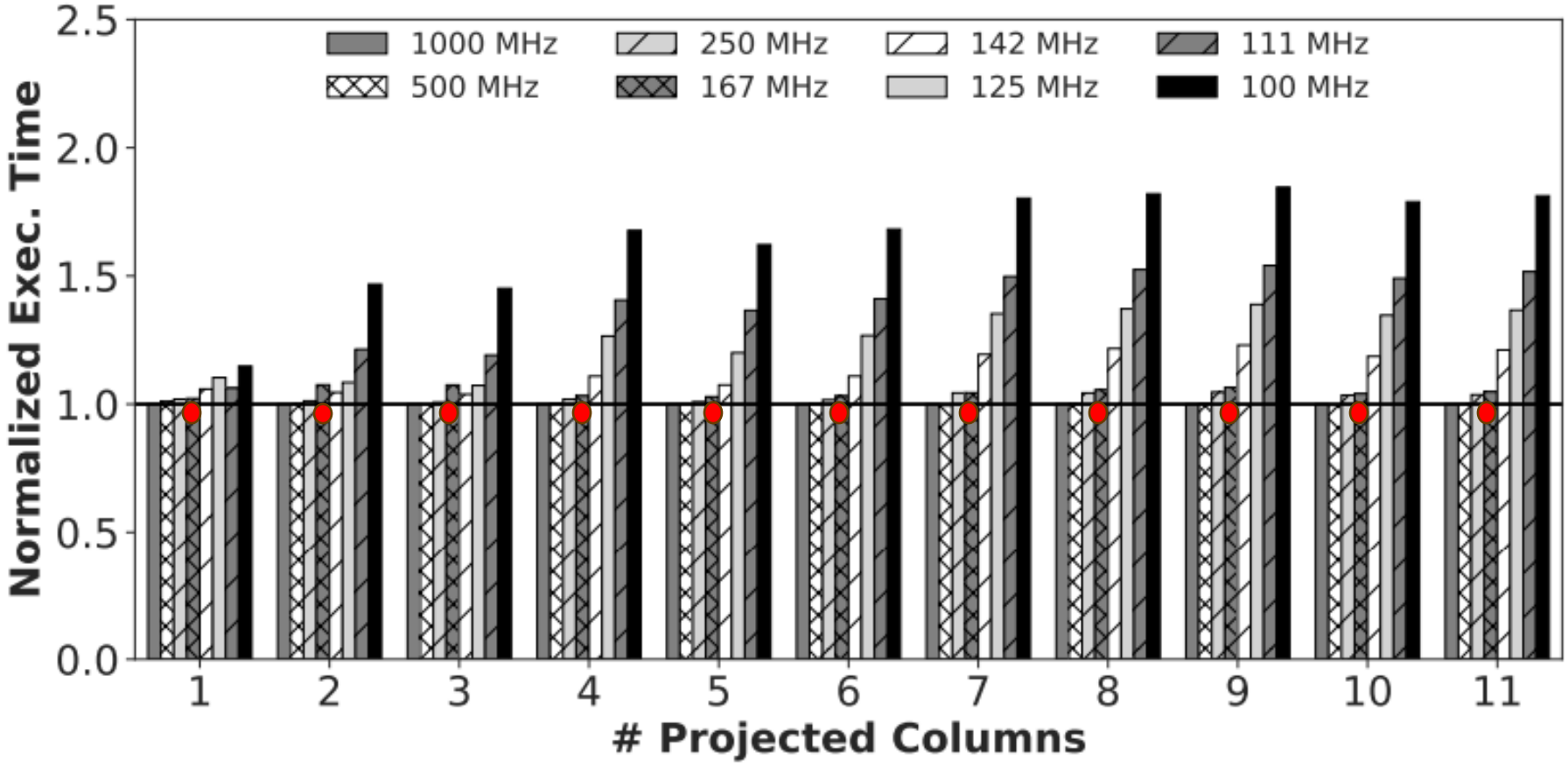
EXP 2: TAKEAWAYS

- Out-of-order core can extract more performance
  - 11 col projection
  - 0.75x out-of-order, 0.89x in-order vs. normalized row-store
- Synergistic effect

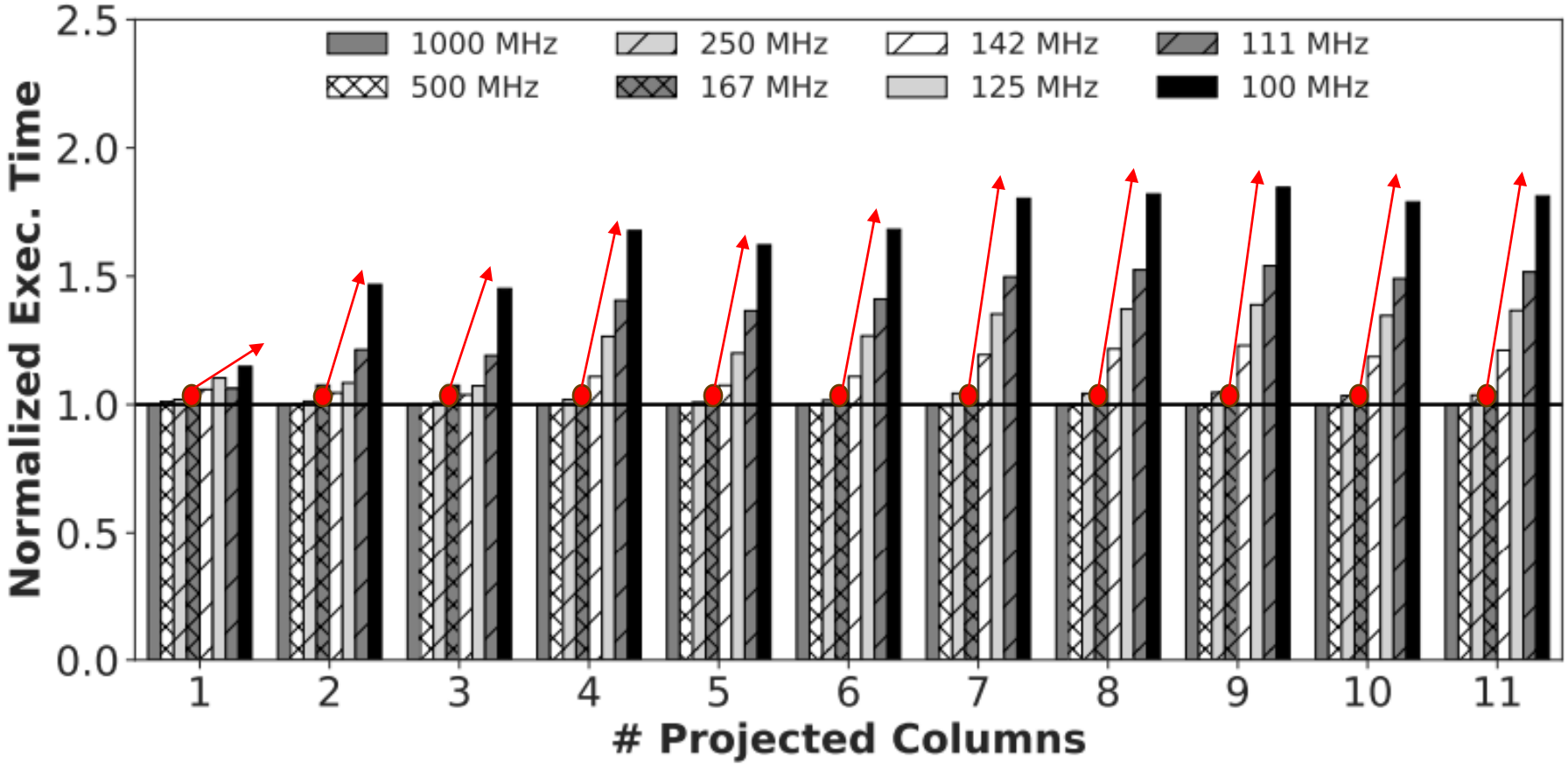
EXP 3: EFFECT OF CLOCK SPEED



EXP 3: EFFECT OF CLOCK SPEED



EXP 3: EFFECT OF CLOCK SPEED



EXP 3: TAKEAWAYS

- Prefetching can hide a significant amount of latency
- Critical threshold when latency **cannot** be hidden
- Clock speed limitation (100 MHz) of PS-PL system could have decreased performance

CONCLUSION

- Contributions
  - Re-implemented and improved RME
  - Validated on-the-fly data transformation on wide range of platform
  - Micro-architectural exploration and performance analysis
- Findings
  - Pre-fetching is able to hide RME latency
  - Higher performance core acts synergistically with RME