An Effective Approach to Processing in DRAM

Jinho Lee, **Kiyoung Choi**, and Jung Ho Ahn

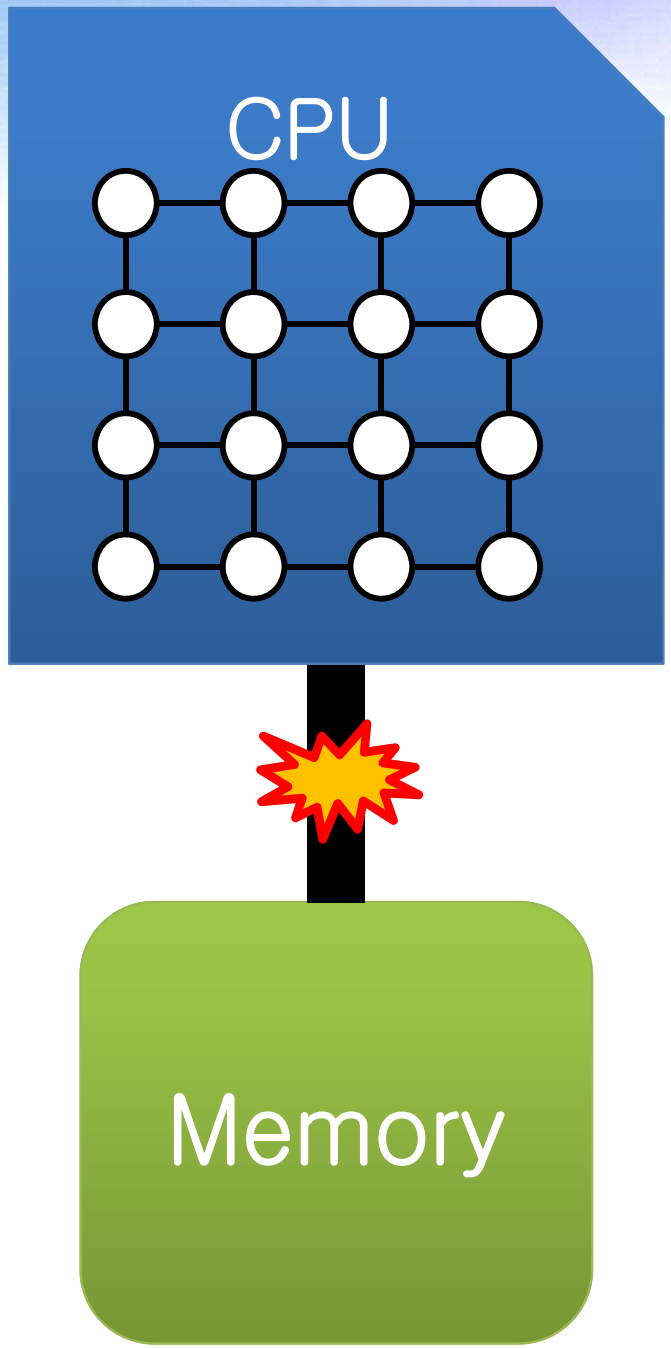
Seoul National University

Outline

- Introduction
- Our Approach
- Buffered Compare Architecture
- Evaluation
- Summary

Introduction – Memory Wall

- The number of cores in a chip is increasing
- The memory bandwidth is not as much…
  --> “memory wall” problem
- Emerging big data applications require even more bandwidth
- In reality, much of the bandwidth is being wasted!

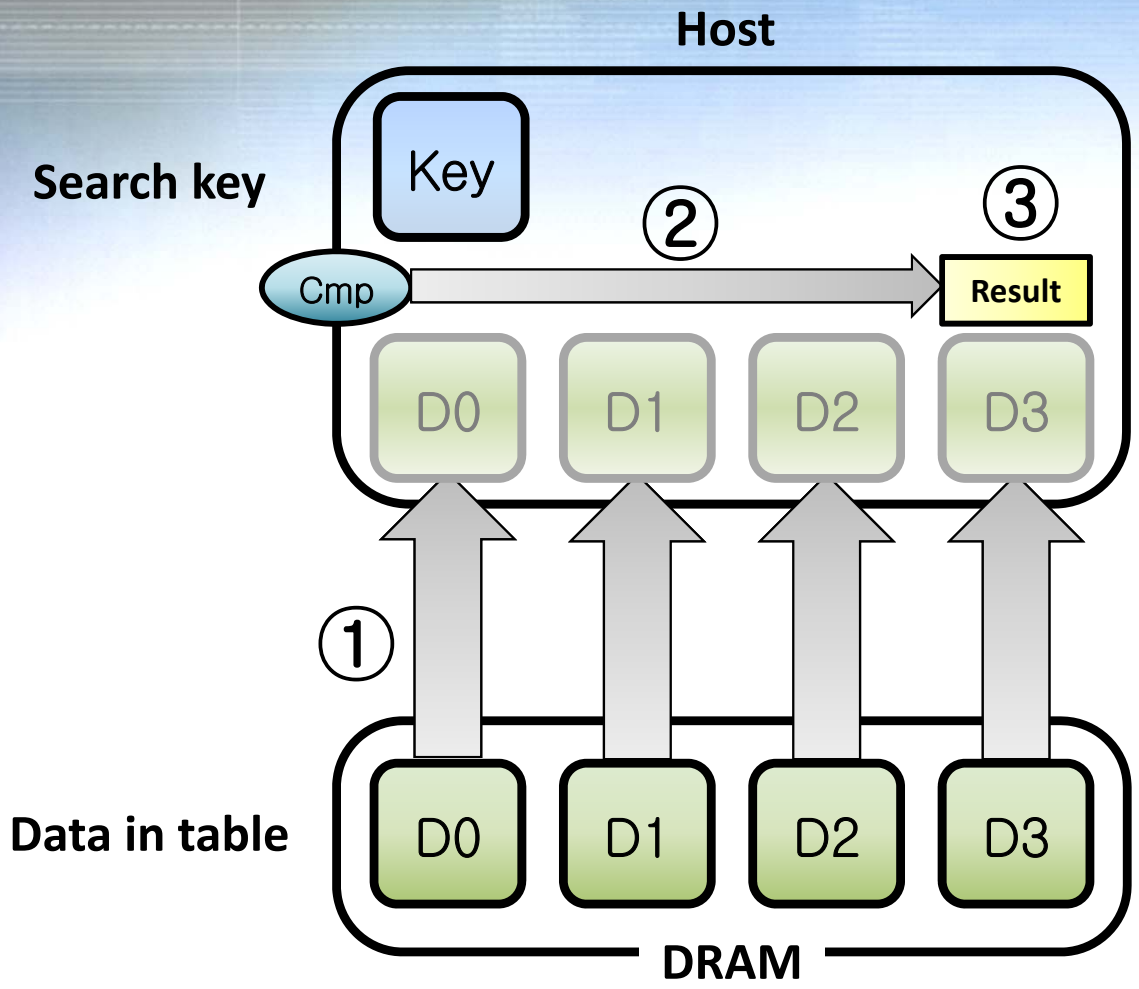


Introduction – Table Scan

- Which items are made of wood?
- Which items are heavier than 5kg?

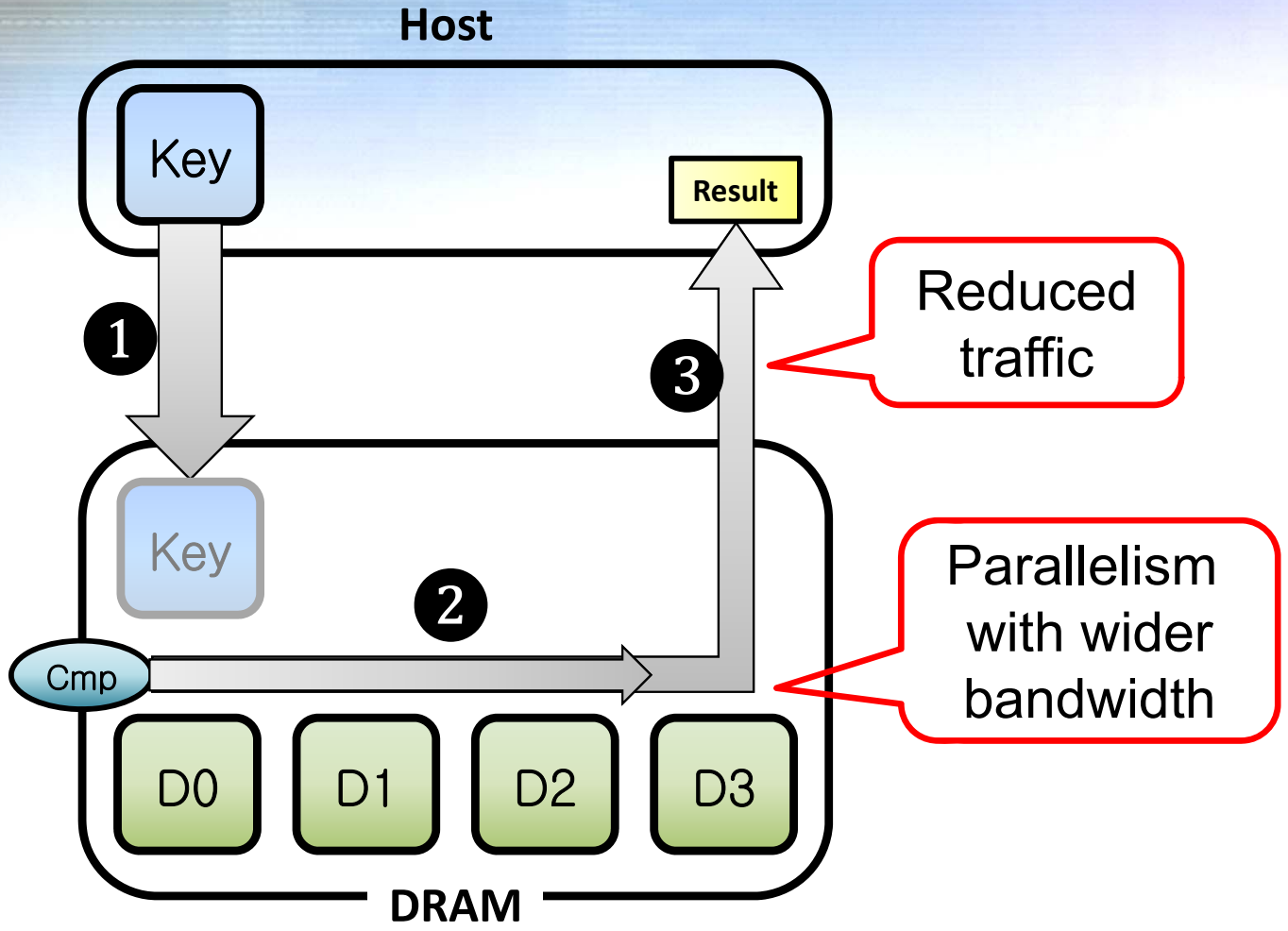
Item#	Material	Weight
A	Wood	10kg
B	Metal	1.5kg
C	Metal	7kg
D	Stone	3kg
E	Wood	2kg
…		

Introduction – Table Scan



- Data are read and the comparisons are done
- We only need the result – waste in bandwidth!

Introduction – Table Scan



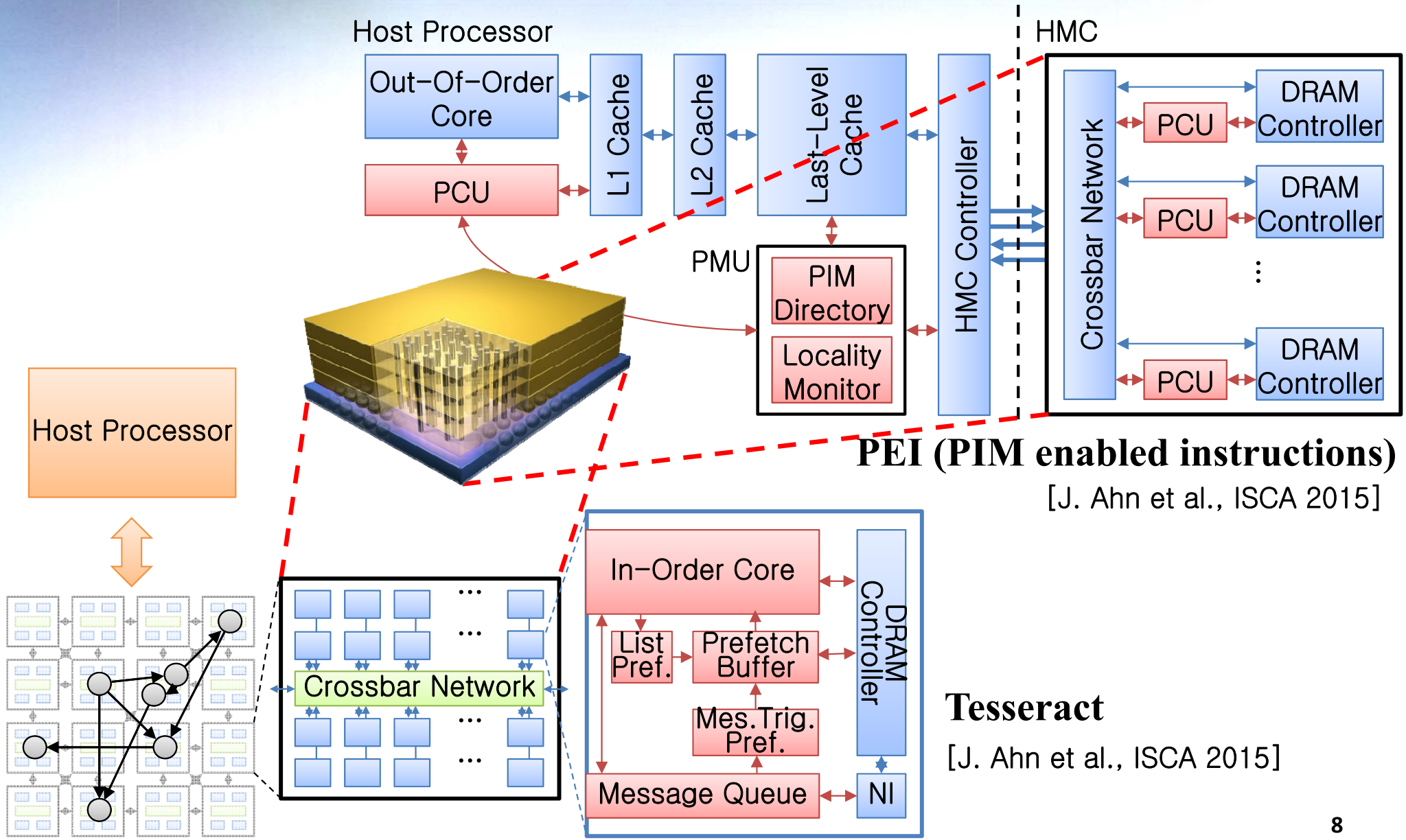
- Do compare within the memory
- Only two transfers needed instead of many
- Essentially a PIM (processing-in-memory) approach

Introduction - PIM

- PIM research was active late 90's ~ early 00's
  - EXECUBE, IRAM, FlexRAM, Smart memory, Yukon, DIVA, etc.
  - Multiple cores in DRAM
  - Hard to integrate --> not successful
- Re-gaining interests due to
  - Big data workloads
  - Limited improvement of memory bandwidth
  - 3D stacked memory (HMC, HBM, etc.) enables integration of cores

Introduction - PIM

- PIM with 3D stacked memory



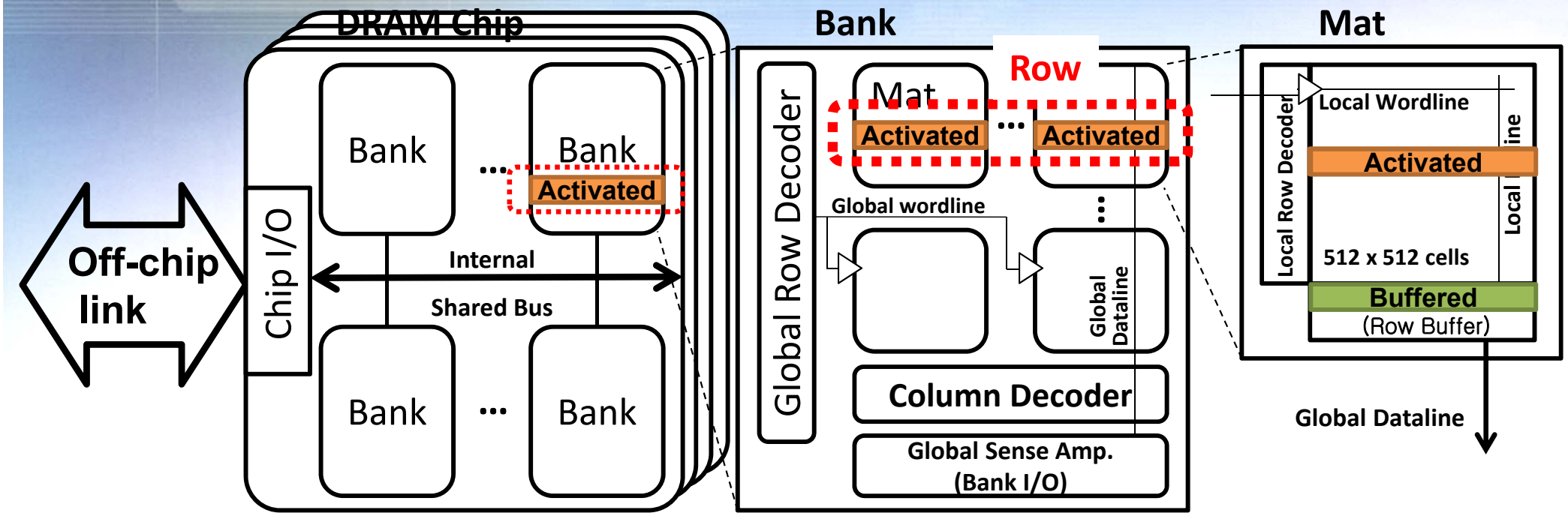
PEI (PIM enabled instructions)

[J. Ahn et al., ISCA 2015]

Tesseract

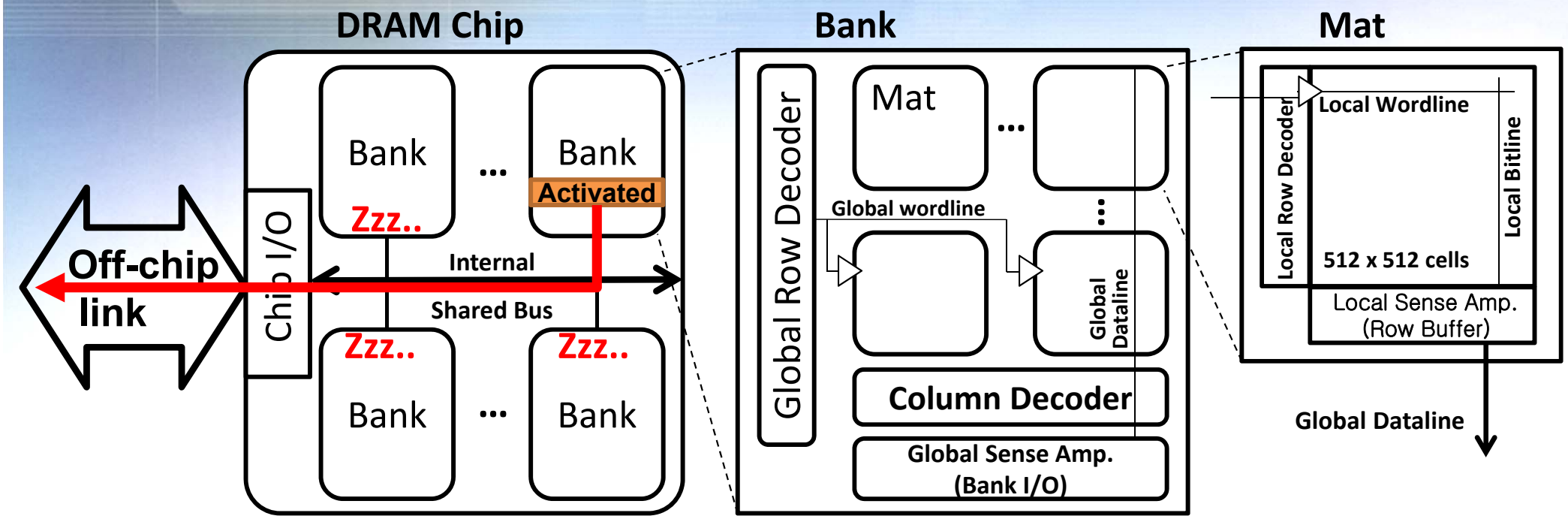
[J. Ahn et al., ISCA 2015]

Our Approach - DRAM Architecture & Motivation



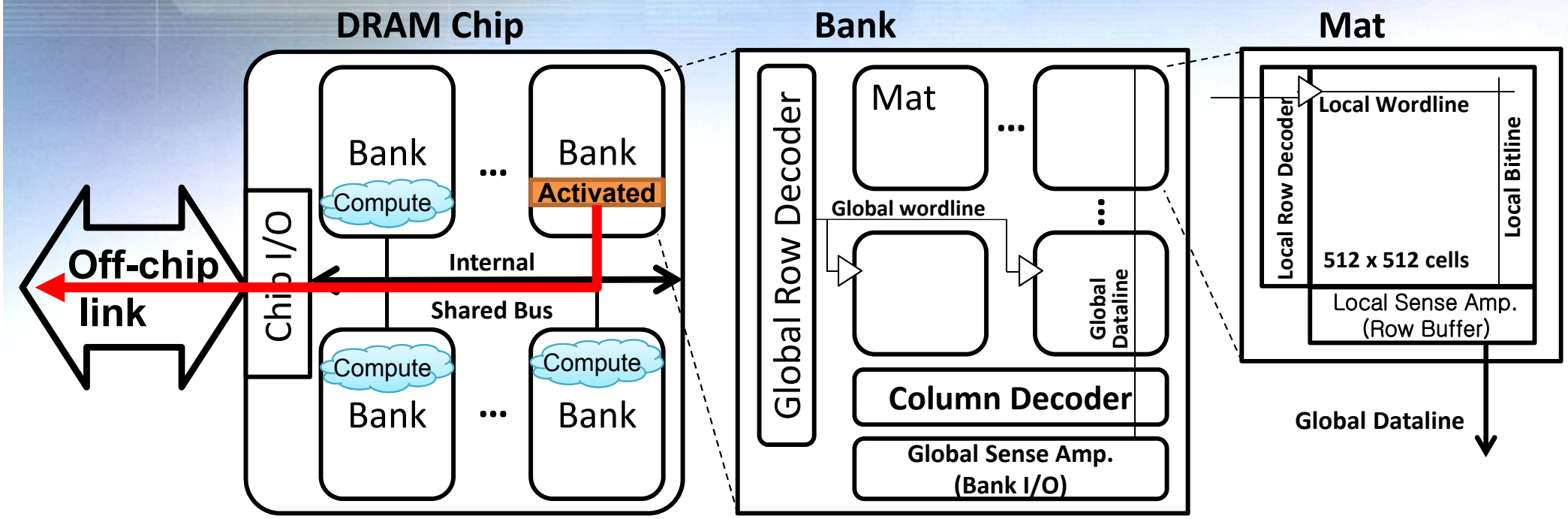
- A single chip is comprised of 8-16 banks
- When accessing data, a row in a bank is "activated" and stored in a row buffer
- A cache line (64B) is fetched in one burst

Our Approach - DRAM Architecture & Motivation



- Multiple banks are used for interleaving since activating a row takes long time
- One bank can fill up the bandwidth for the off-chip link
- Thus we have 8X-16X internal bandwidth, most of which is wasted

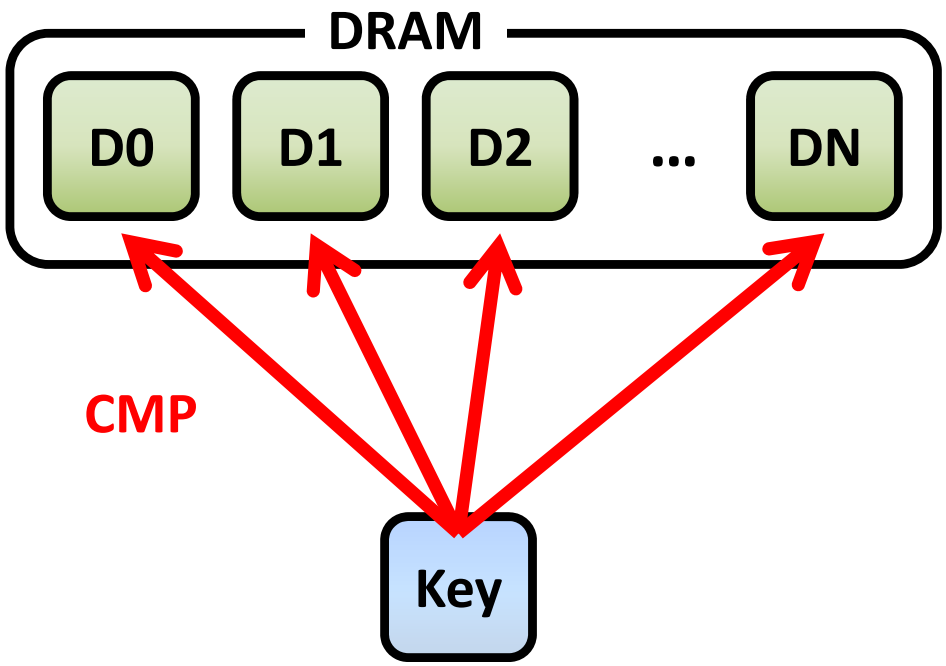
Our Approach - DRAM Architecture & Motivation



- Compute inside each bank to utilize the excess bandwidth

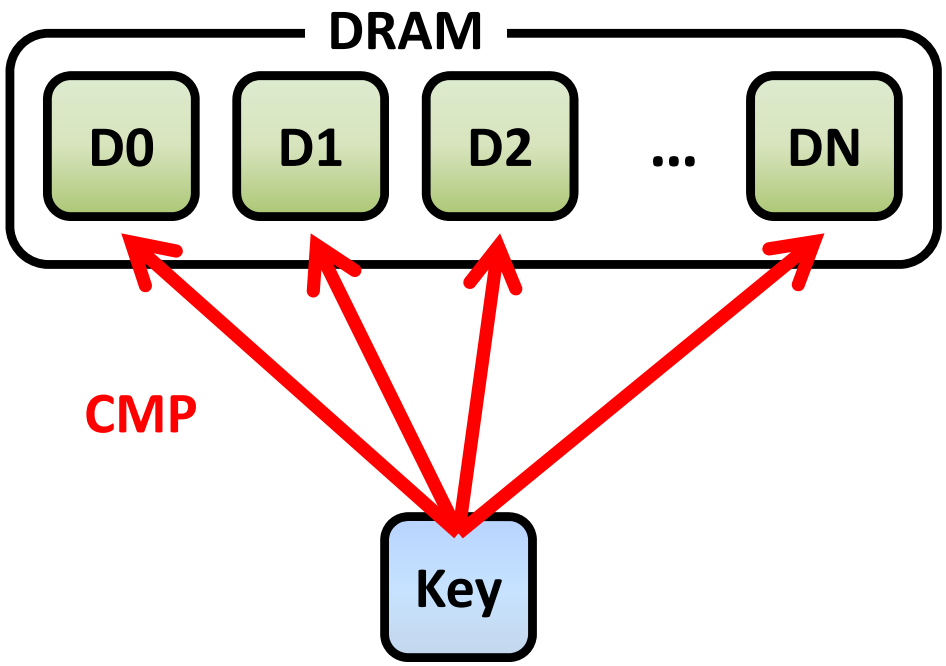
Our Approach - What to compute with PIM?

- We focus only on ‘compare-n-op’ pattern over a long range of data



Our Approach - What to compute with PIM?

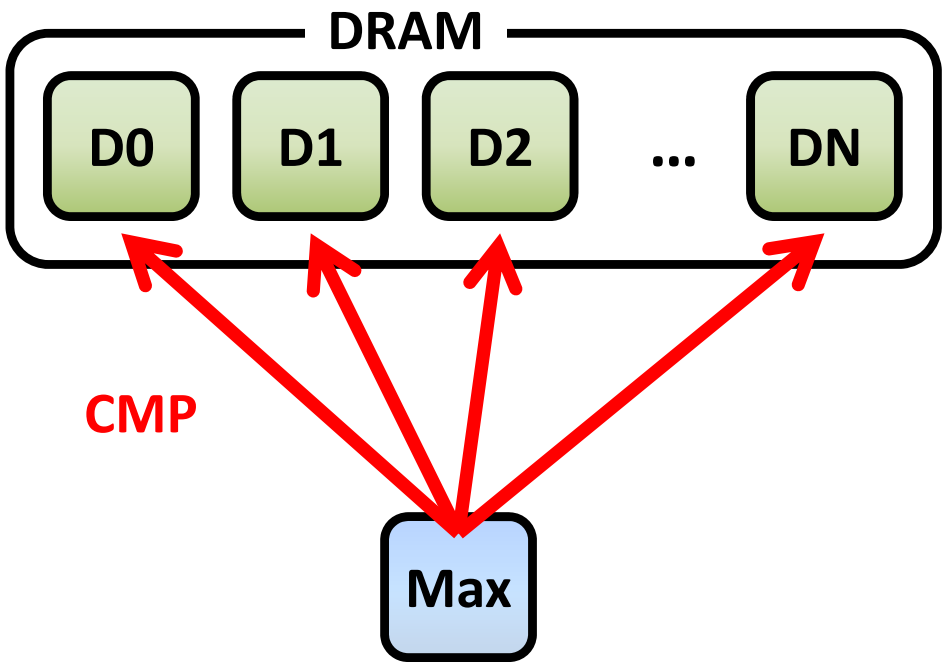
- Compare-n-read
  - Returns the match results for each item



Result: (=, <, =, … , >)

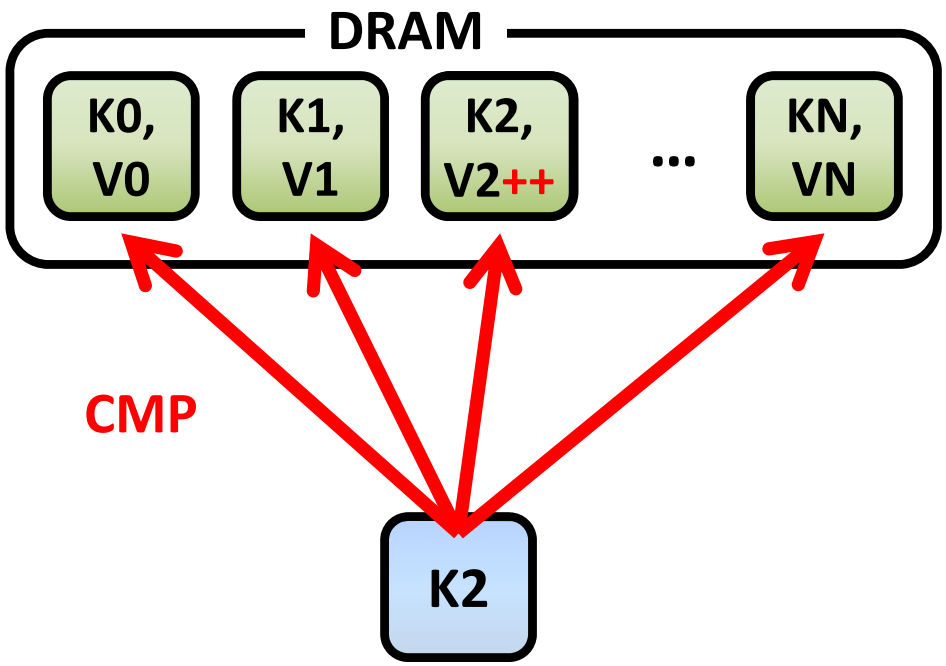
Our Approach - What to compute with PIM?

- Compare-n-select
  - Returns the min/max among each item

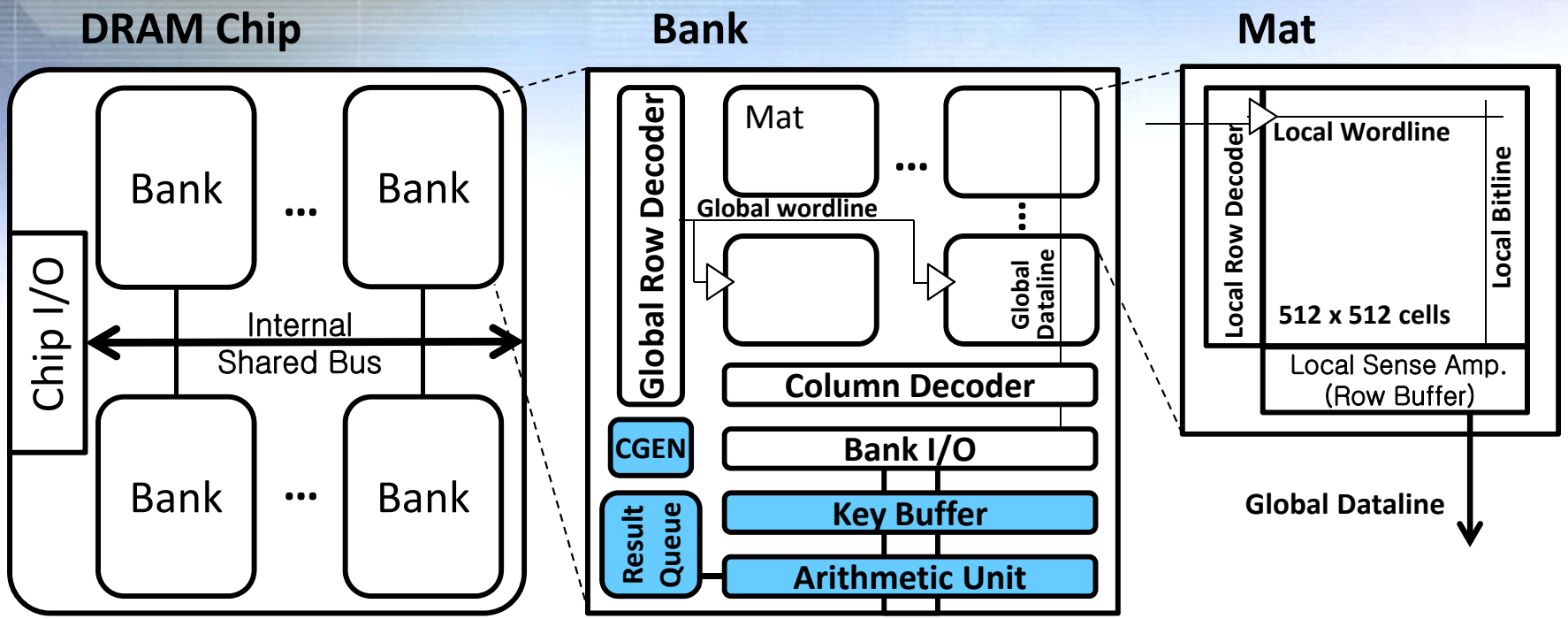


Our Approach - What to compute with PIM?

- Compare-n-increment
  - Increments matching items

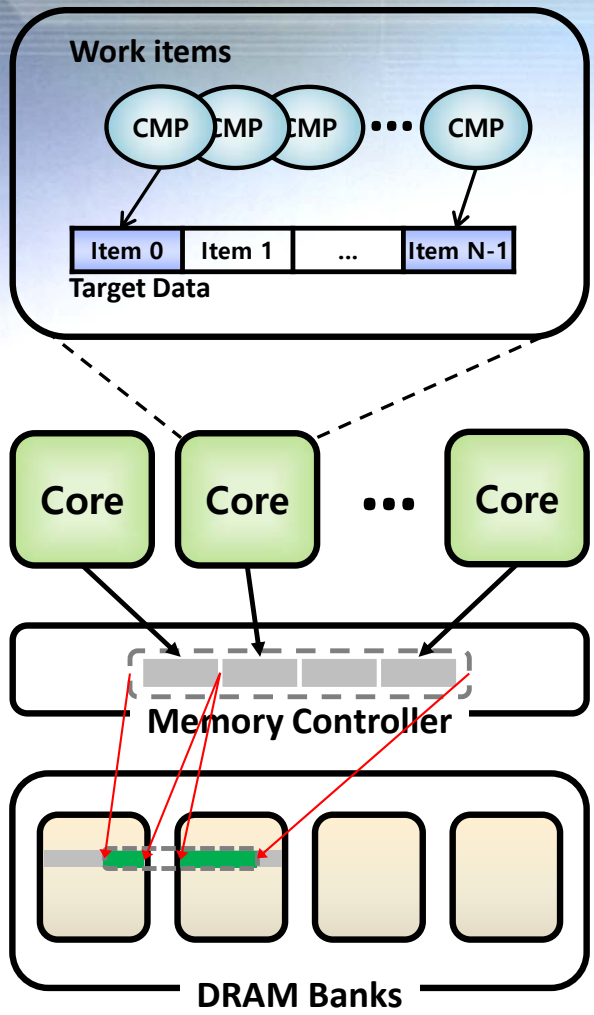


Buffered Compare Architecture



- **Key buffer**: Holds a value written by the processor
- **Arithmetic unit**: Performs computation (cmp, add, etc.) using Bank I/O and Key buffer as operands
- **Result queue**: Stores compare results
- **CGEN**: Repeats the bank-local commands
- The datapath is 64 bits wide
- 0.53% overhead in DRAM area

Buffered Compare Architecture - Programming Model



SW code

```
__kernel search(keys[], skey, d[]){
  int id  = get_global_id(0)
  if (keys[id] == skey)
     d[id] = 1
}
```

Instruction

```
BC_cmp_read(skey, keys, N)
…
```

DRAM cmd

```
CMP_RD(skey, addr, range)
```

- OpenCL based programming model
- Programmers need not be aware of DRAM parameters (page size, number of banks, …)

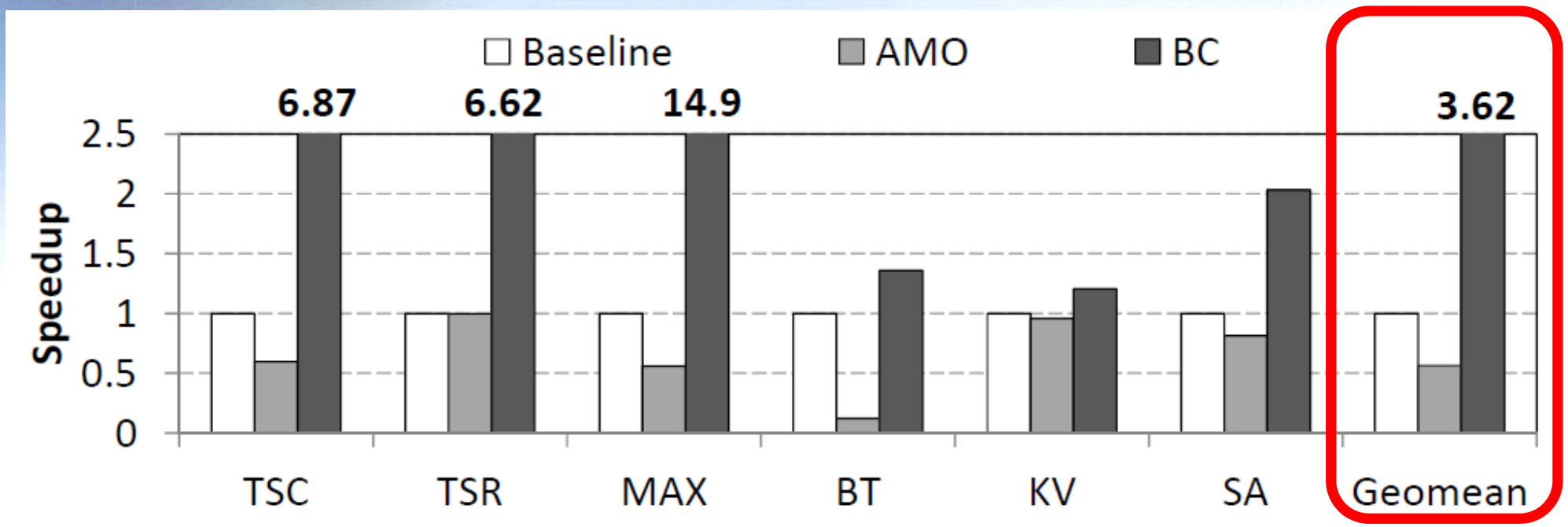
Evaluation - Setup

- McSimA+ simulator
- Processor
  - 22nm, 16 OoO cores running at 3GHz
  - 16KB private L1
  - 32MB S-NUCA L2
  - Directory-based MESI coherence
- Memory
  - 28nm
  - DDR4-2000
  - 4 ranks per channel
  - 16 banks per chip
  - PAR-BS (parallelism-aware batch scheduling)

Evaluation - Setup

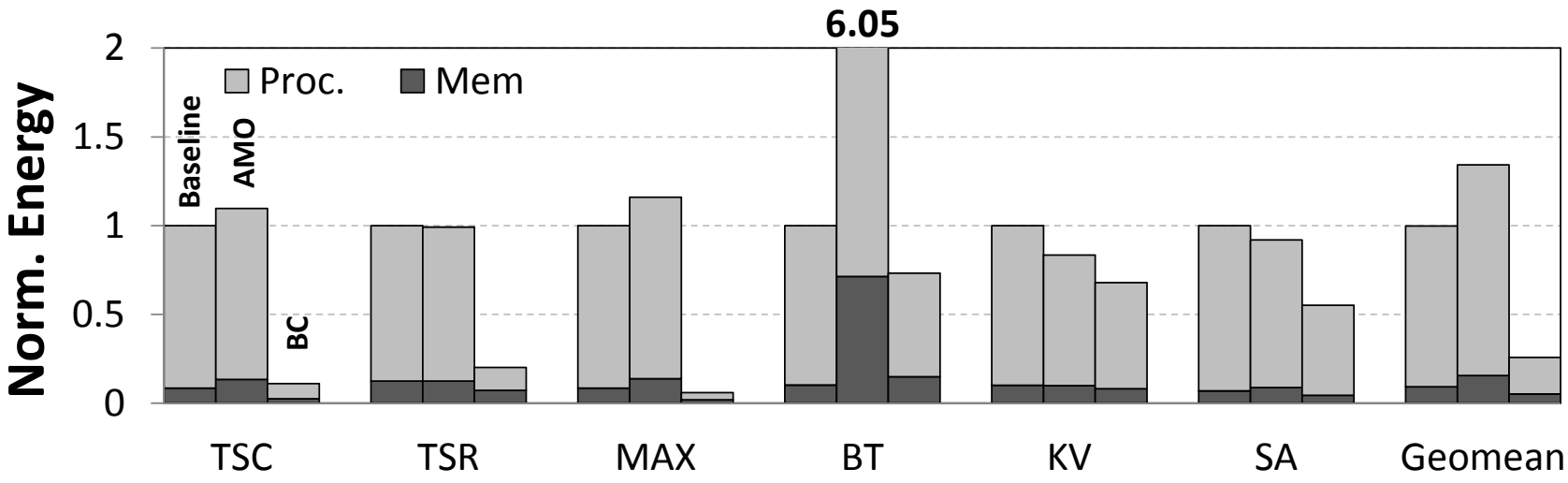
- Six workloads
  - TSC : In-memory linear scan (Column-store)
  - TSR : In-memory linear scan (Row-store)
  - BT : B+ tree traversal (index scan)
  - MAX : MAX aggregation
  - SA : Sequence assembly
  - KV : Key-value store
- BC was evaluated against baseline and AMO (Active Memory Operation)

Evaluation - Speedup



- BC performs 3.62 times better than the baseline

Evaluation – Energy Reduction



- Energy consumption reduced by 73.3% on average
  - Proc: 77.2%
  - Mem: 43.9%

Summary

- We proposed buffered compare, a processing-in-memory approach to utilizing internal bandwidth of DRAM
  - Minimal overhead to the DRAM area
  - Less invasive to existing DDR protocols
  - 3.62X speedup and 73.3% energy reduction

- Limitations
  - Utilization of cache
  - Utilization of critical-word-first policy
  - When using x4 devices, only up to 32bits are supported for the operands