PROCESSING IN MEMORY

Mahdi Nazm Bojnordi

Assistant Professor

School of Computing

University of Utah

Overview

- Upcoming deadlines
  - March 27th: sign up for your student paper presentation
  - Prepare for exactly 20m talk followed by 5m Q&A

Presenters	Date
Kohl, Meher, Shirley	March 29
Karl, Anirban, Chandrasekhar	April 3
Suryanarayanan, Tim, Arjun	April 5
Pranav, Goverdhan, Yomi	April 10
Munzer, Manikanth, Amandeep	April 12

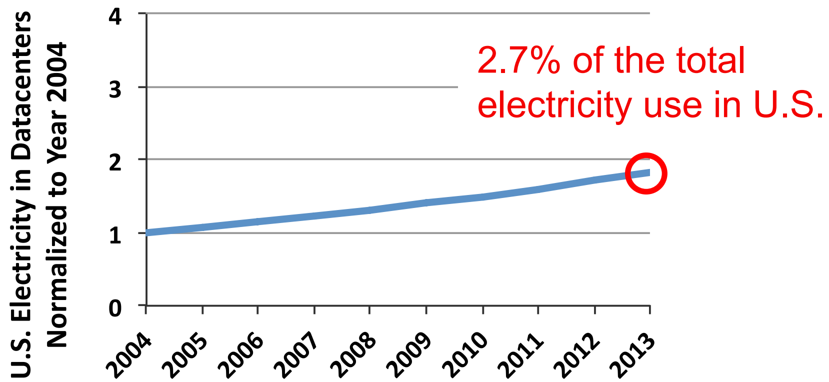
- Solidify the title of your paper presentation by email!

Overview

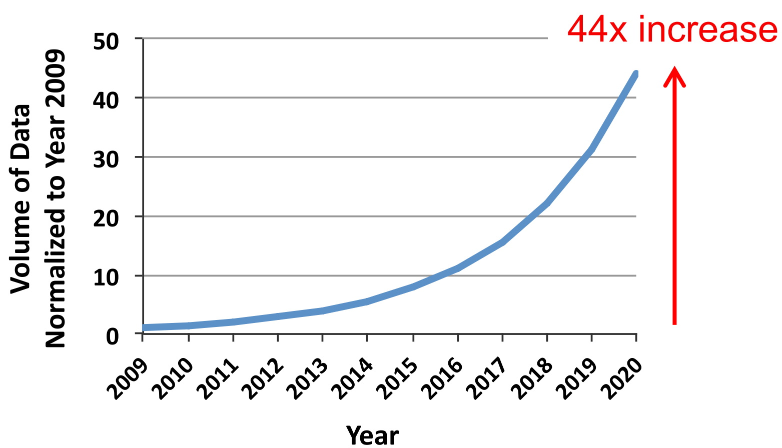
- This lecture
  - Trends in data processing
  - Trends in technology
  - Intelligent RAM
  - The Raw microprocessor
  - Processing on DIMM

Trends in Data Processing

- The electricity used by U.S. data centers increases at an annual rate of 7%
- Since 2009, 41% more data is created each year



[DOE, 2015][1]

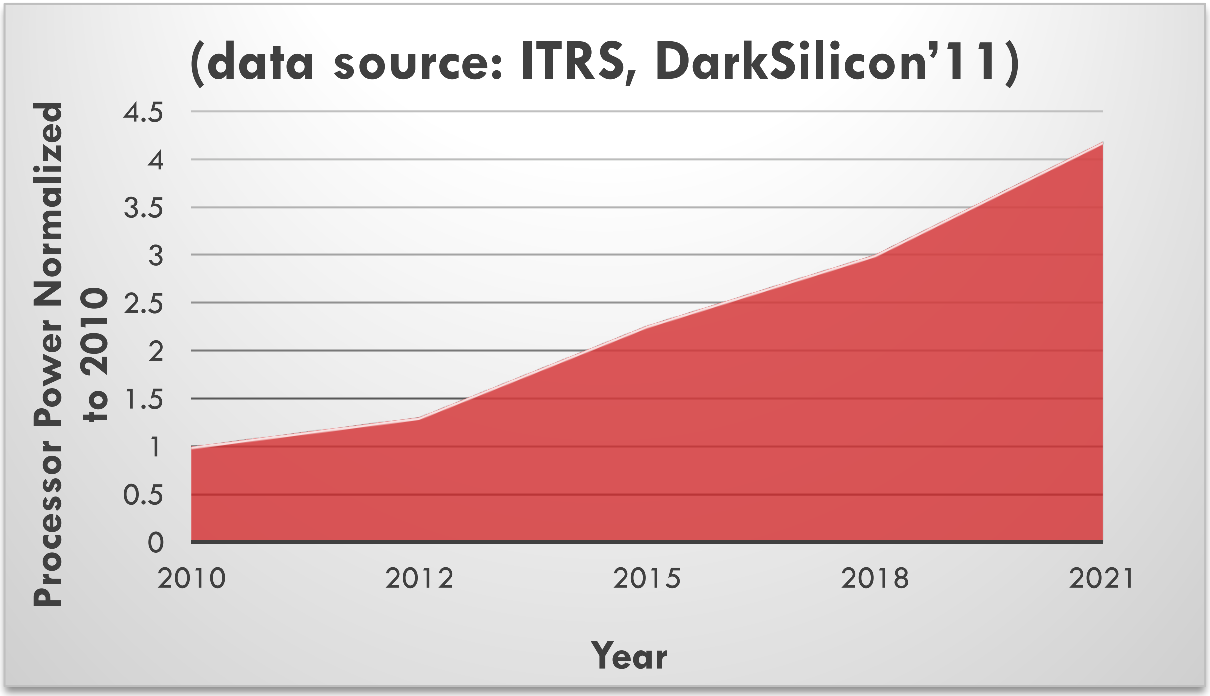


[J. E. Short *et al.*, 2011][2]

1. DOE, "Potential for data center efficiency improvement", 2015
2. J. E. Short *et al.*, "How much information? 2010 report on enterprise server information", 2011

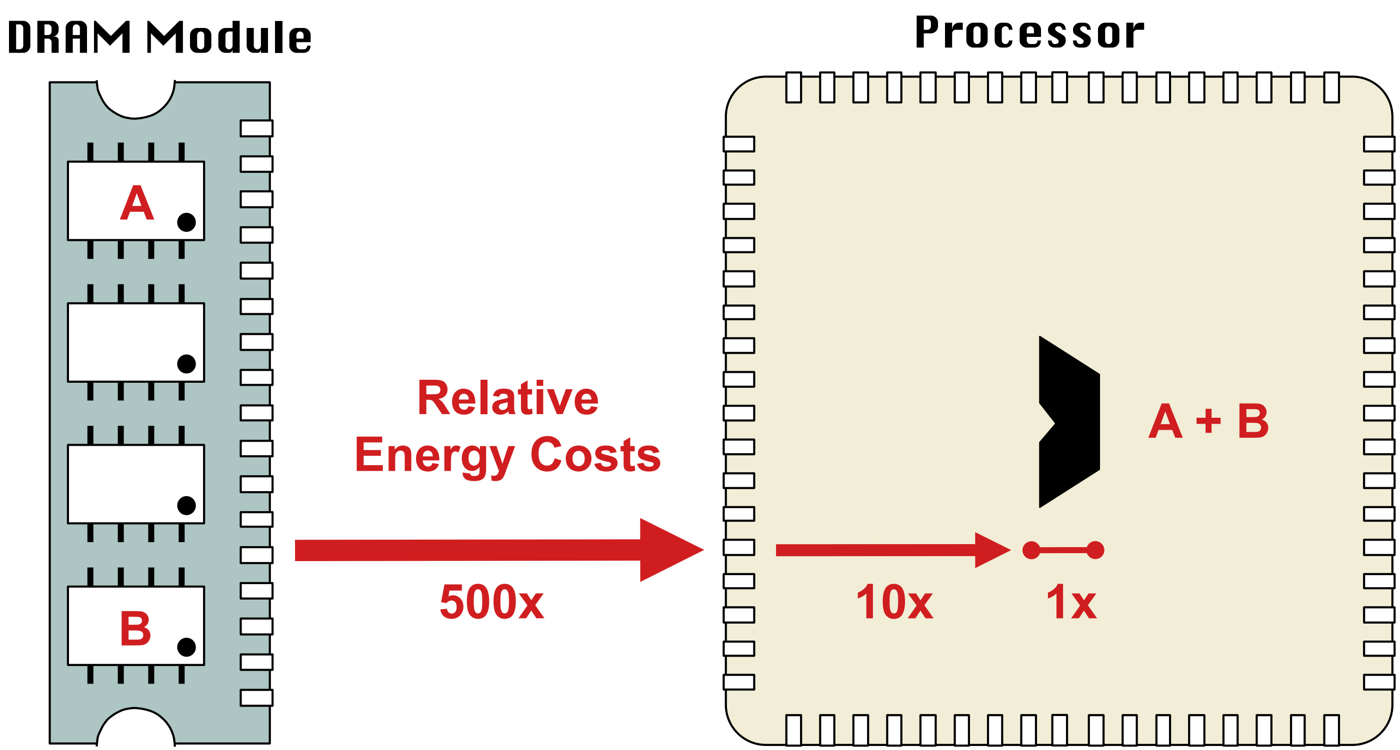
Energy and Power Trends

- Power consumption is increasing significantly



The Cost of Data Movement

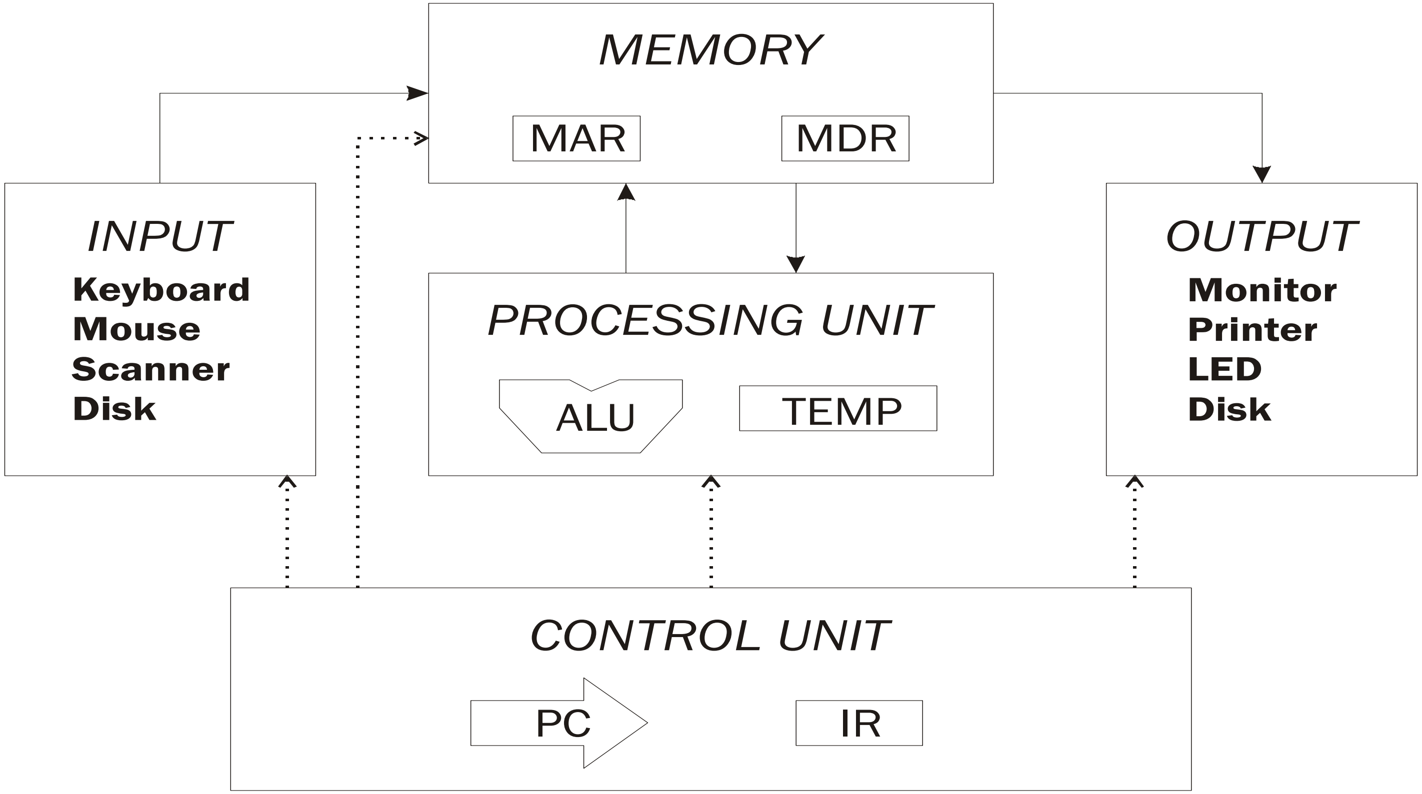
- Data movement is the primary contributor to energy dissipation in nanometer ICs.



Source: NVidia

Computational Models

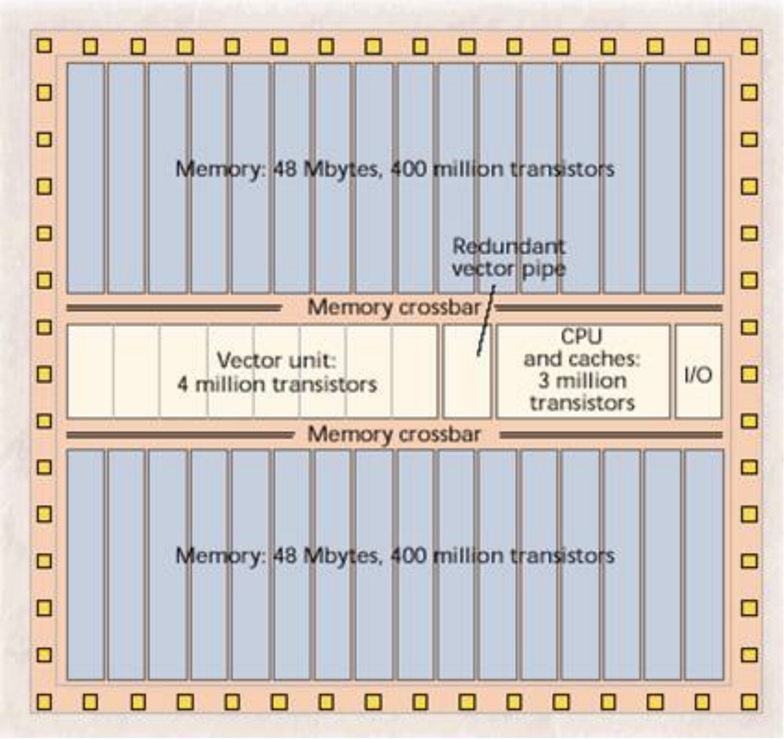
- Von Neumann machine

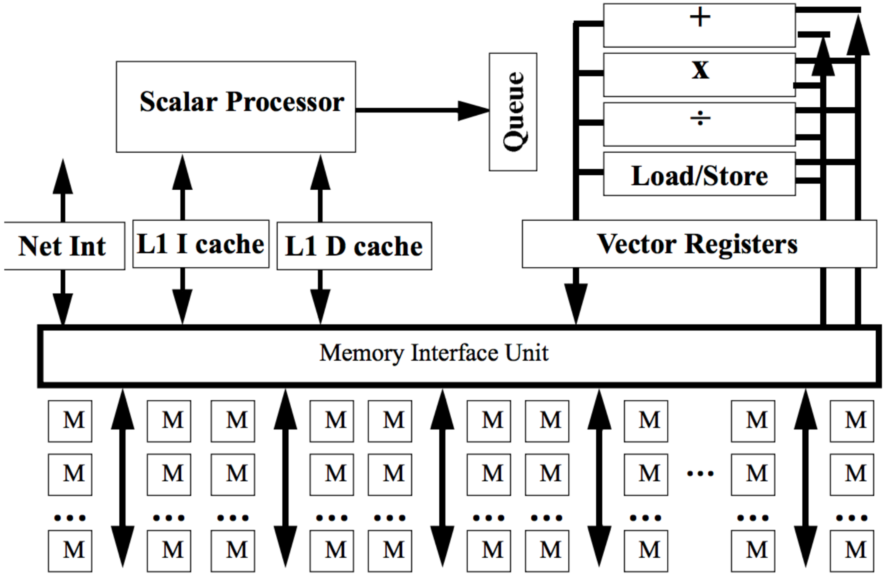


Past Attempts

Intelligent RAM (IRAM)

- A non Von Neumann model
  - Unifying processing and memory into a single chip





[Micro'97]

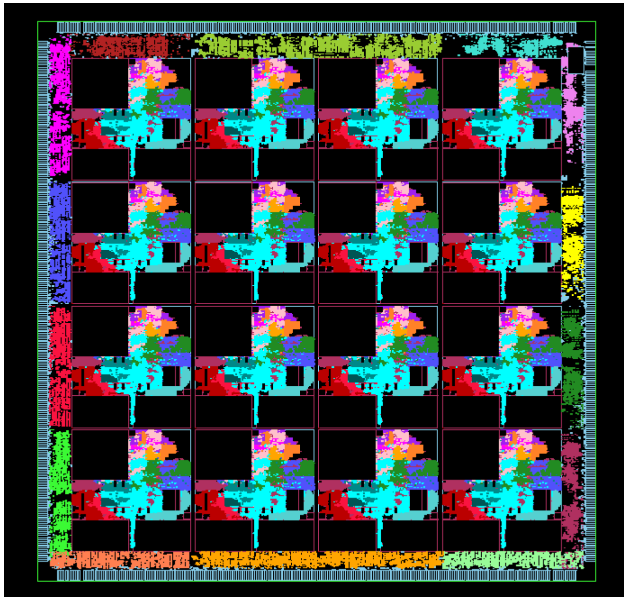
Intelligent RAM (IRAM)

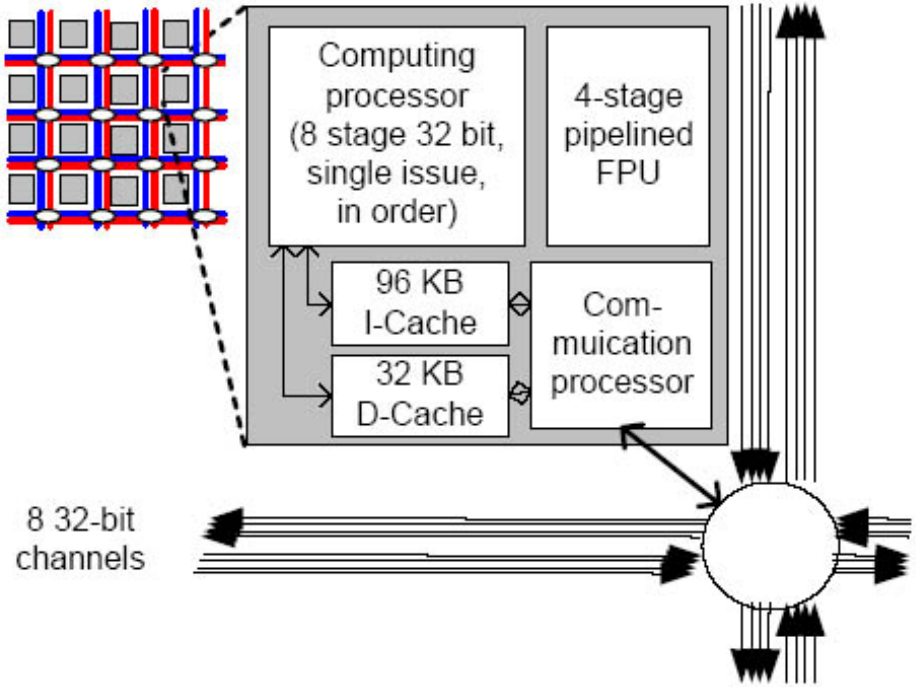
- Merging a microprocessor and DRAM on the same chip
  - Performance
    - reduce latency by 5~10
    - Increase bandwidth by 50~100
  - Energy efficiency
    - Save at 2~4
  - Cost
    - Remove off-chip memory and reduce board area
- IRAM is limited by amount of memory on Chip
- Potential of network computer
- Change the nature of semiconductor industry

[Micro'97]

The RAW Processor

- A scalable 32 bit fabric for general purpose and embedded computing



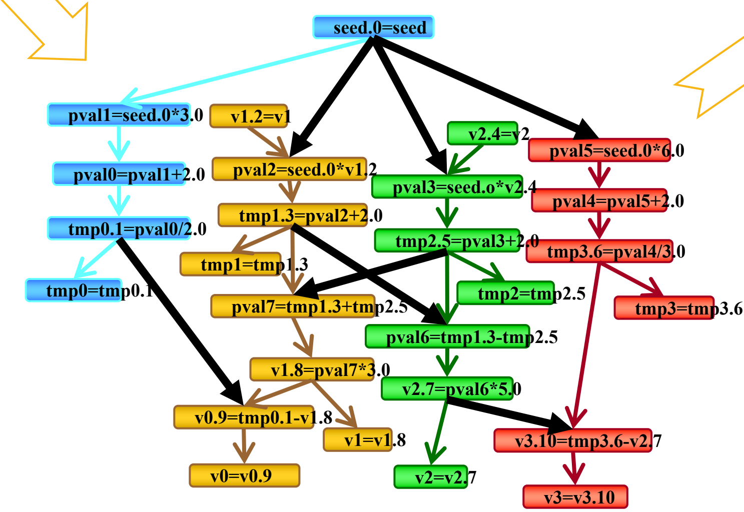


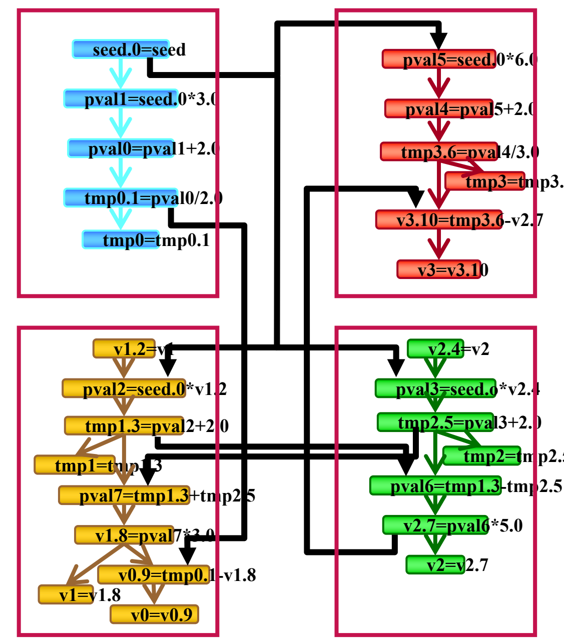
[Micro'04]

The RAW Processor

- It requires complex code generation

tmp0 = (seed*3+2)/2
tmp1 = seed*v1+2
tmp2 = seed*v2 + 2
tmp3 = (seed*6+2)/3
v2 = (tmp1 - tmp3)*5
v1 = (tmp1 + tmp2)*3
v0 = tmp0 - v1
v3 = tmp3 - v2

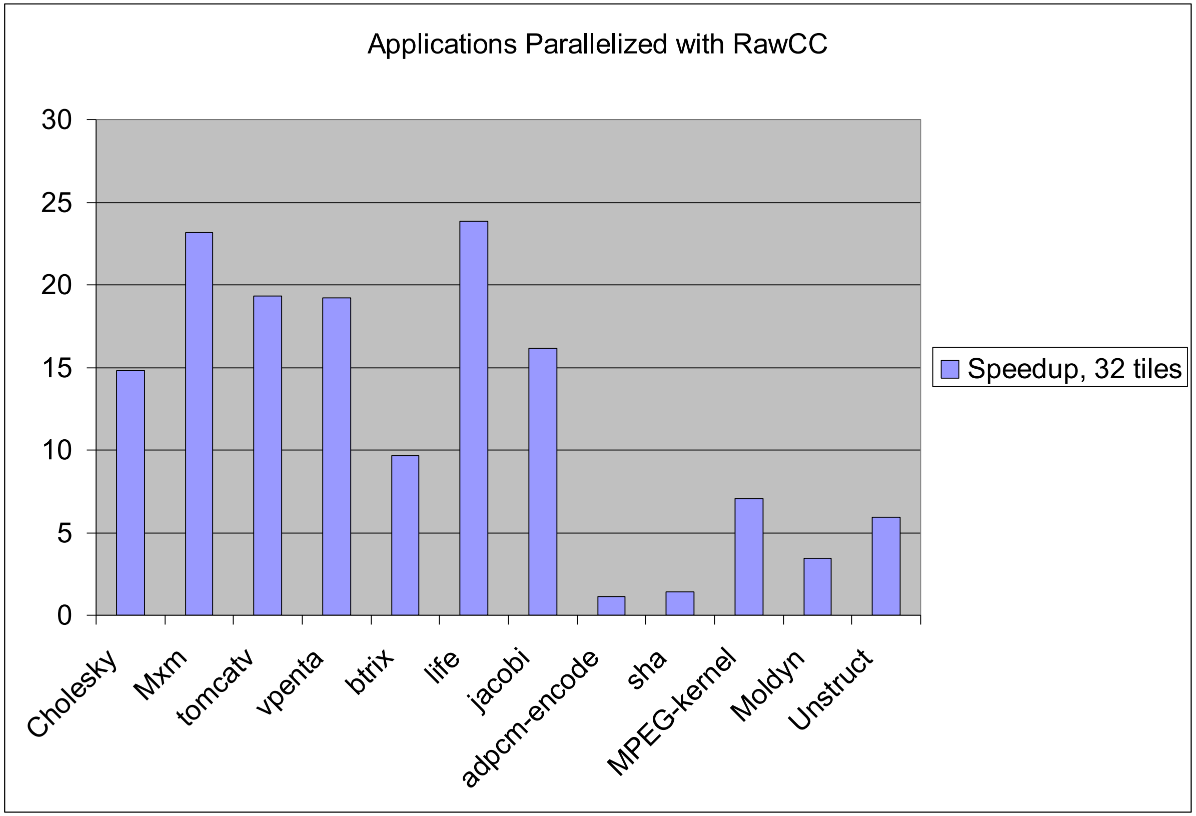




[Micro'04]

The RAW Processor

- 16 Tiles; 2048 KB SRAM On-chip

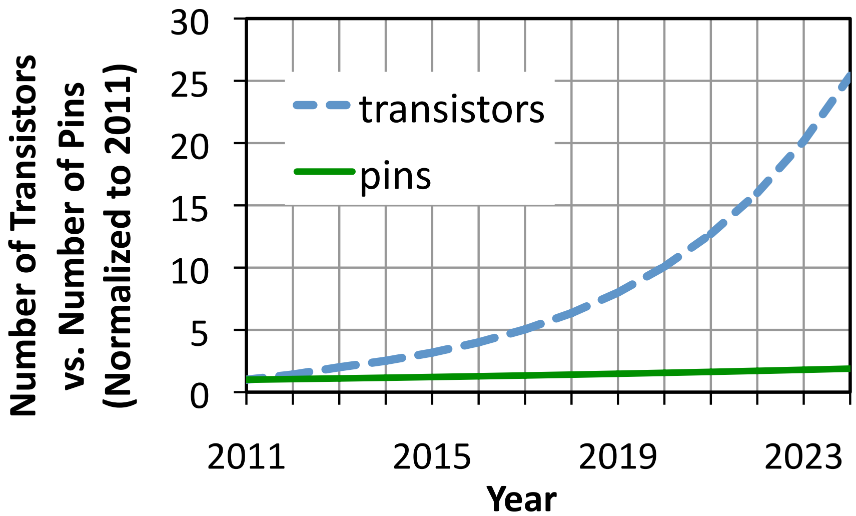


[Micro'04]

Current and Future

Power and Bandwidth Challenges to Scaling

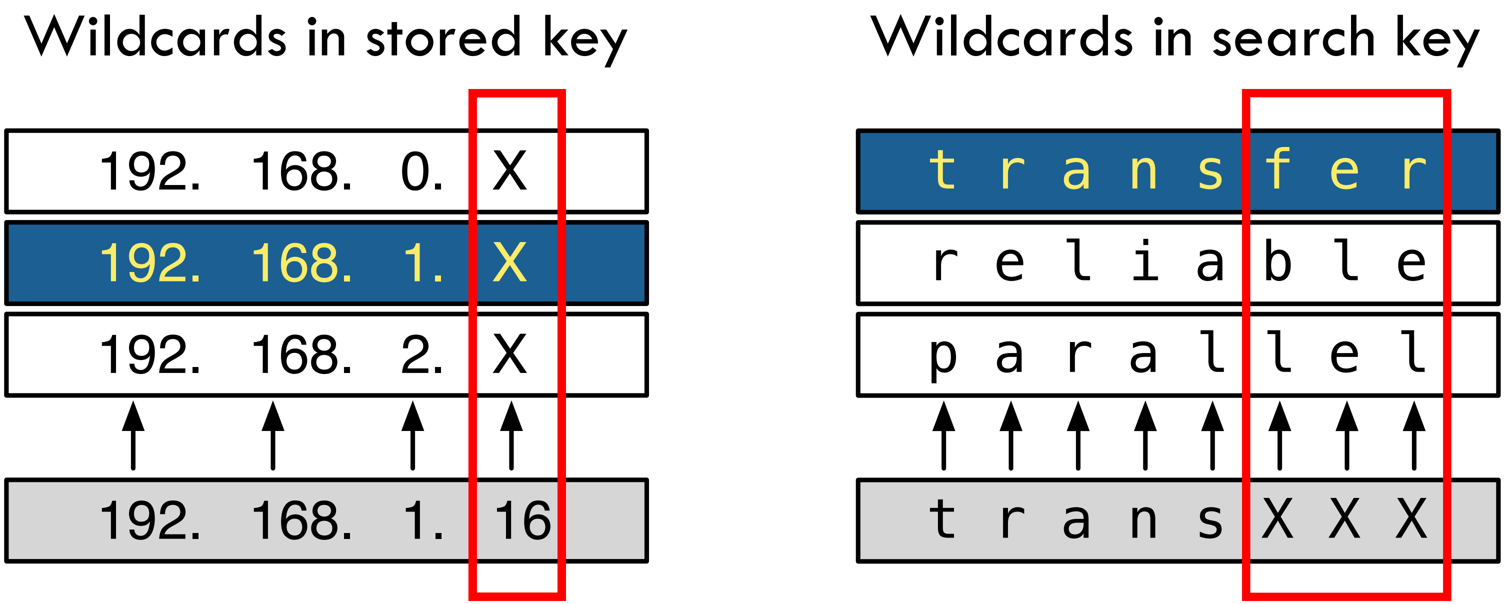
- Transistor density doubles every two years
  - Power efficiency does not scale proportionally
    - 80% of transistors can be simultaneously active at 22nm
    - 50% projected at 8nm†
  - Number of pins grows by 16% / year‡



[MICRO'11]

Ternary Content Addressable Memory

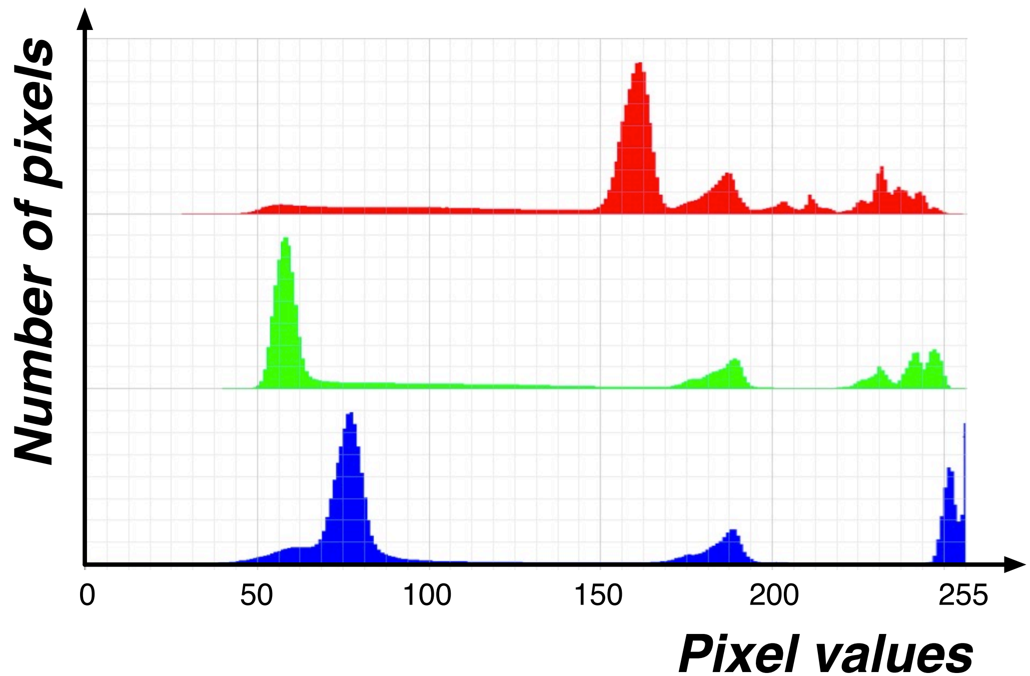
- A TCAM permits storing and searching with wildcards



Example Application: Image Histogram

- Goal: compute pixel value distribution in a digital image

[Phoenix benchmark suite]



Data Intensive Computing

- Data intensive applications are increasingly important
  - Energy and bandwidth hungry
  - Ubiquitous
    - Data mining
    - Machine learning
    - Web search
    - Database management
    - Video and image processing

- Possible solution: associative computing with CAMs

[MICRO'11]

Example Application: Image Histogram

- RAM-based: scan

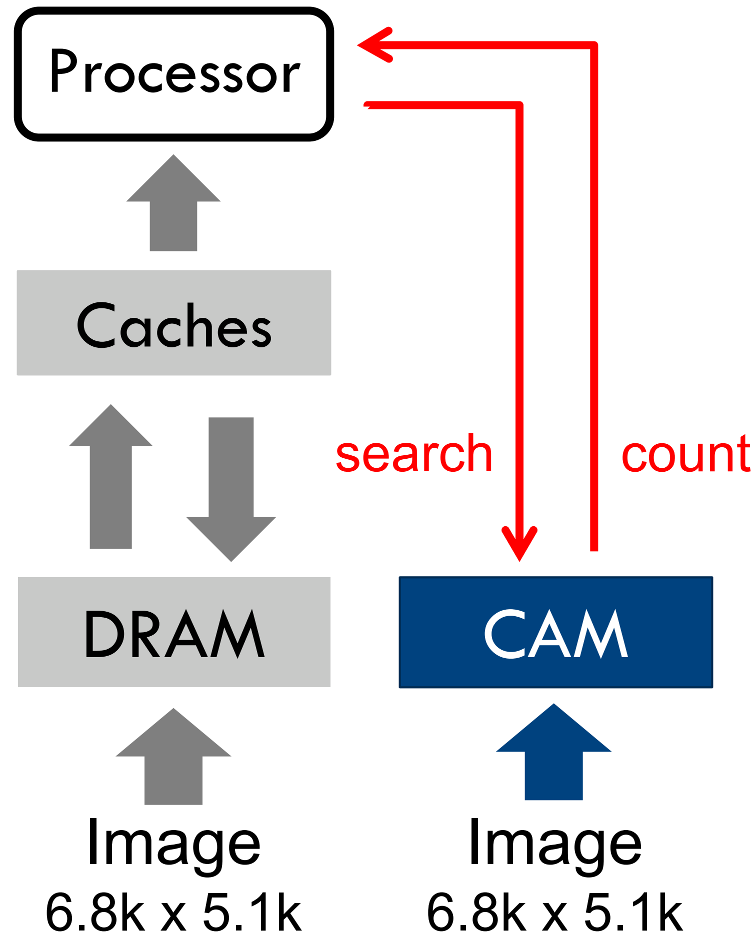
  For 6816 × 5112 pixels

  - ~100 MB reads
  - ~10^8 additions

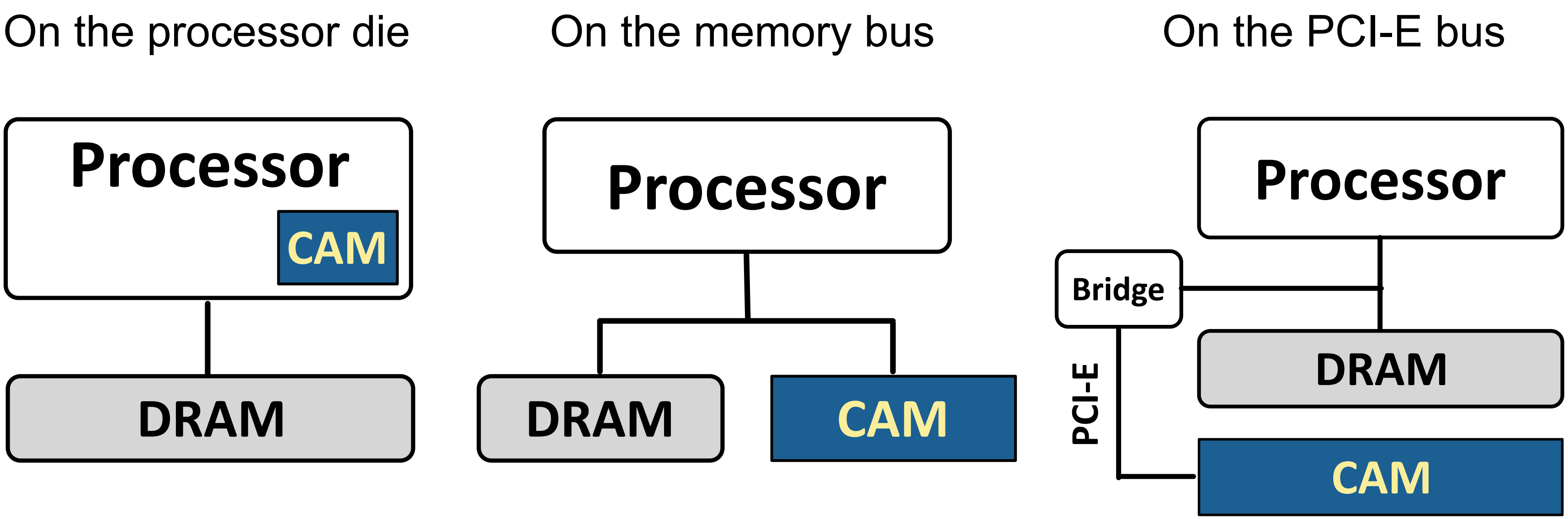
- CAM-based: search

  For the same image

  - 256 × 3 searches
  - 256 x 3 reads



Where in the System Does CAM Belong?

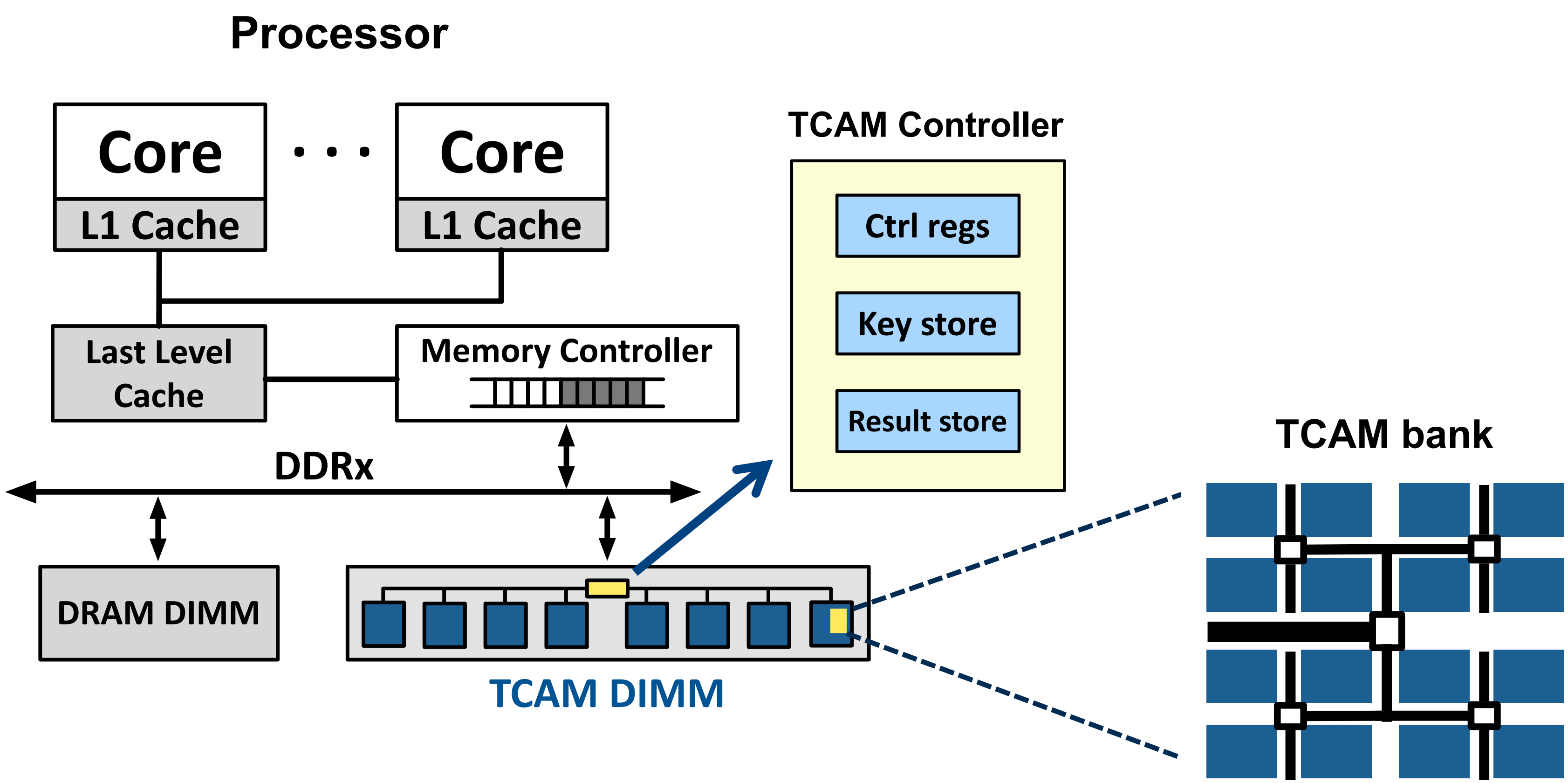


On the processor die	On the memory bus	On the PCI-E bus
– not modular	+ modular	+ modular
+ low latency	+ acceptable latency	– high latency
– limited capacity	+ acceptable capacity	+ high capacity
	✔	

[MICRO'11]

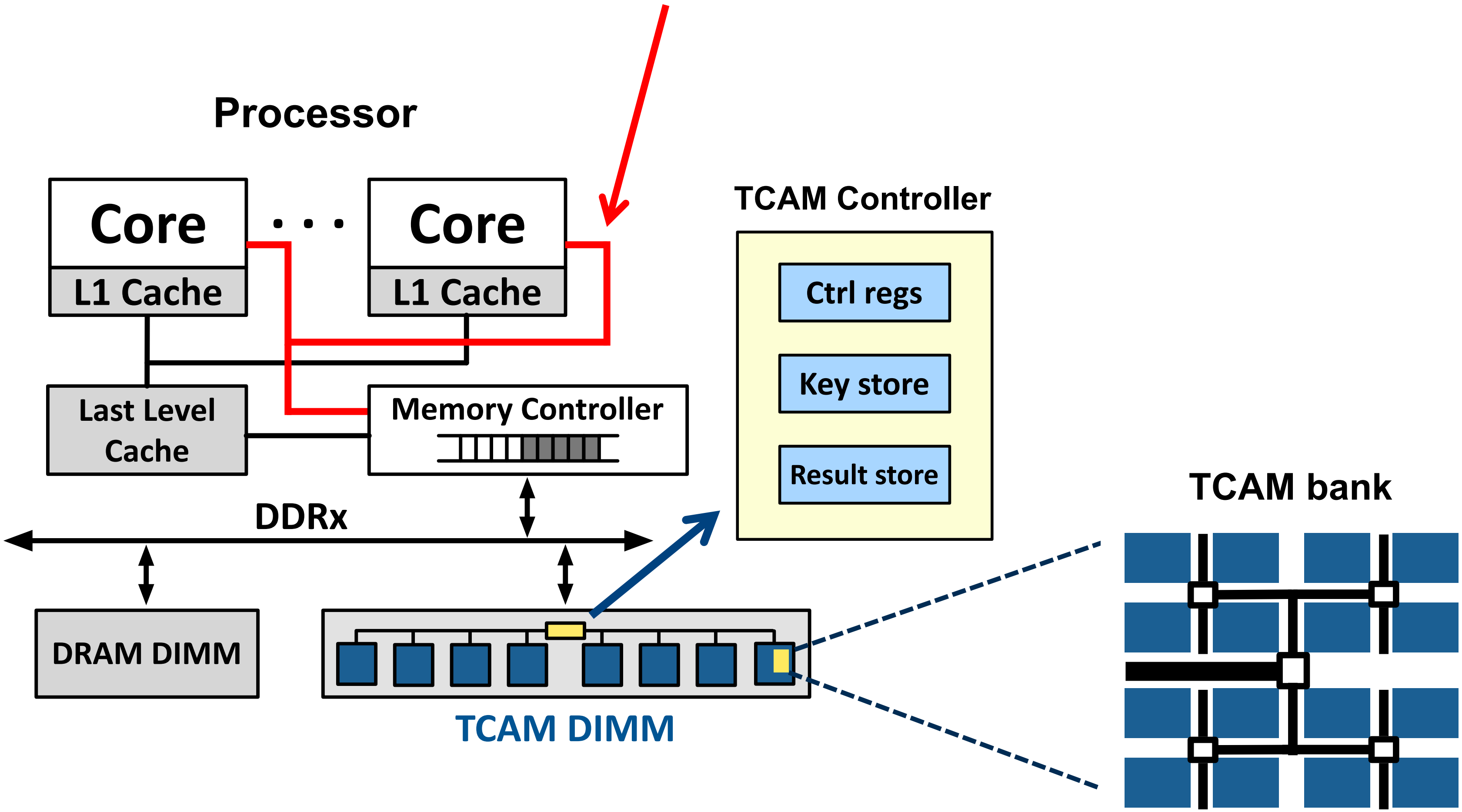
TCAM DIMM [MICRO'11]

- Processor uses uncacheable loads and stores to access TCAM



Processor Interface

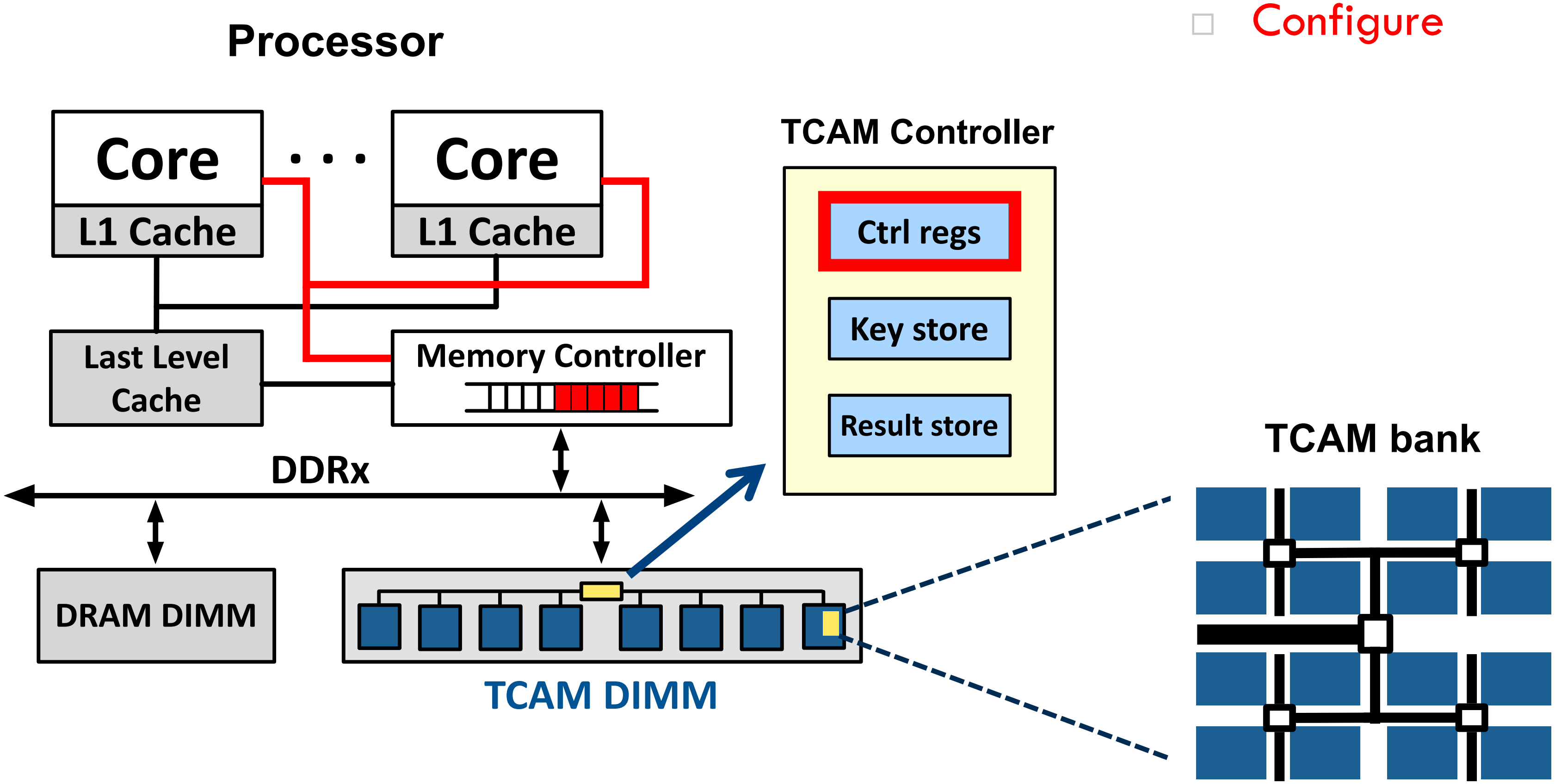
- Processor uses uncacheable loads and stores to access TCAM



Processor Interface

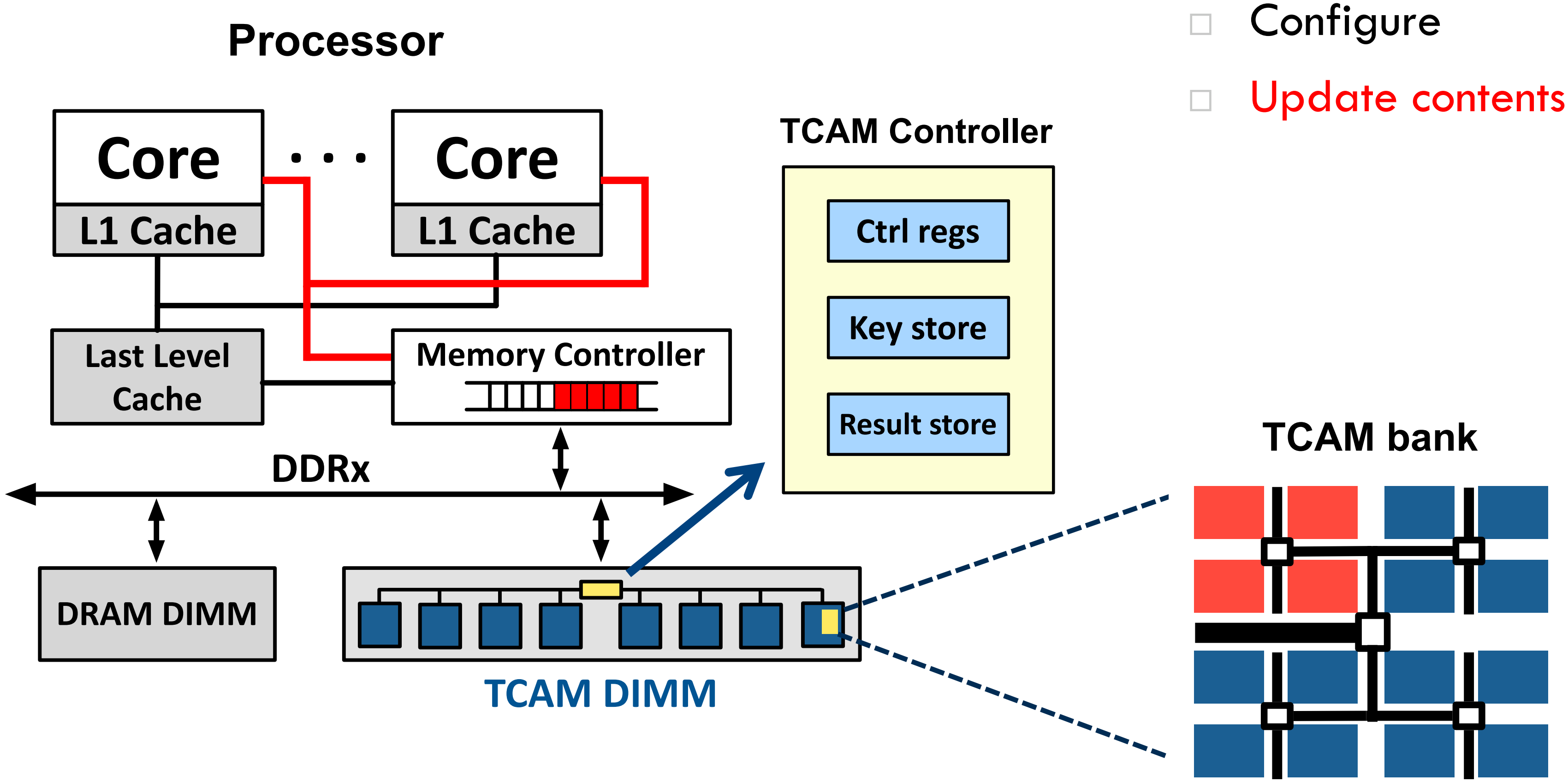
- Processor uses uncacheable loads and stores to access TCAM

- Configure



Processor Interface

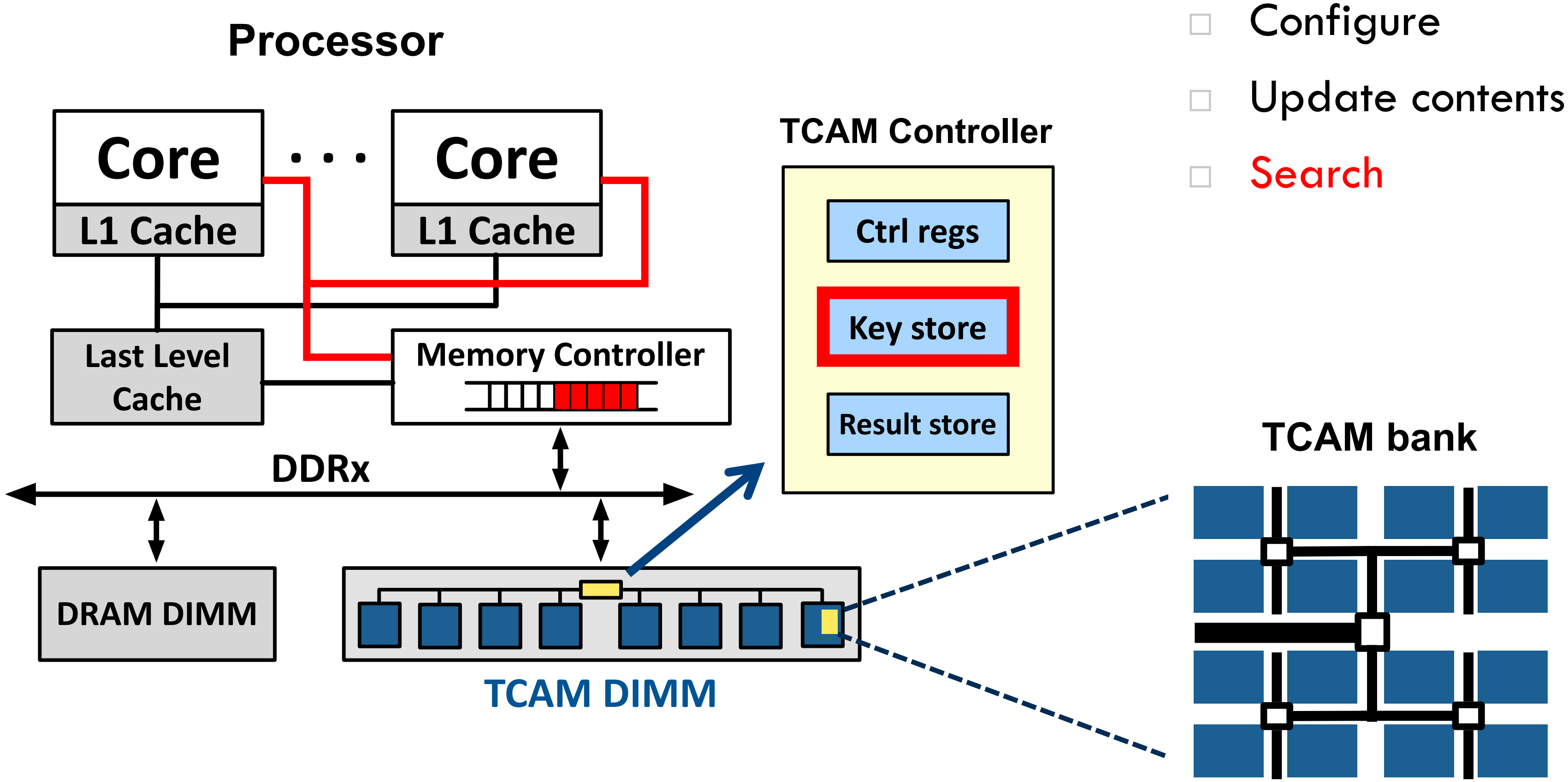
- Processor uses uncacheable loads and stores to access TCAM



- Configure
- Update contents

Processor Interface

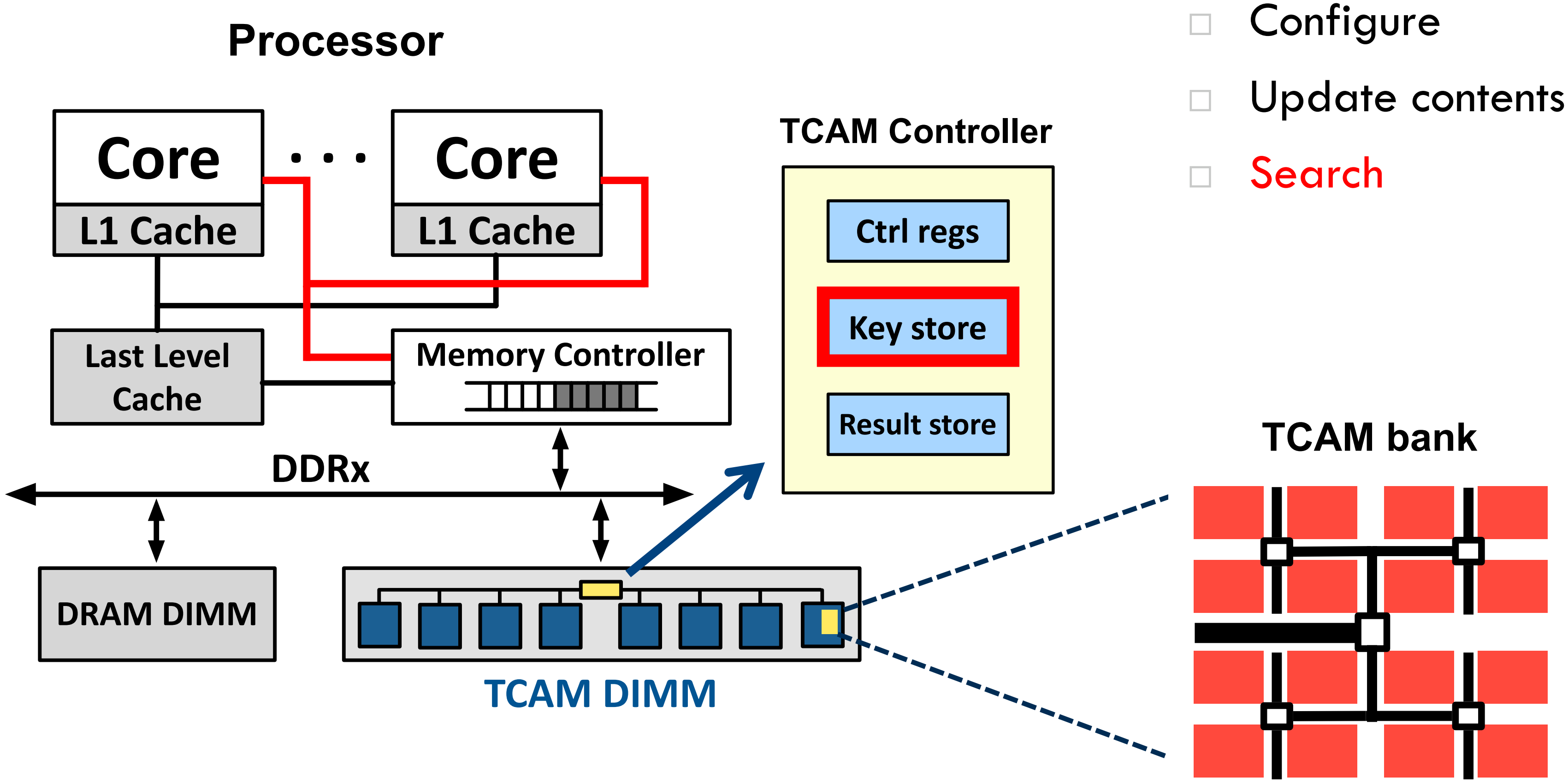
- Processor uses uncacheable loads and stores to access TCAM



- Configure
- Update contents
- Search

Processor Interface

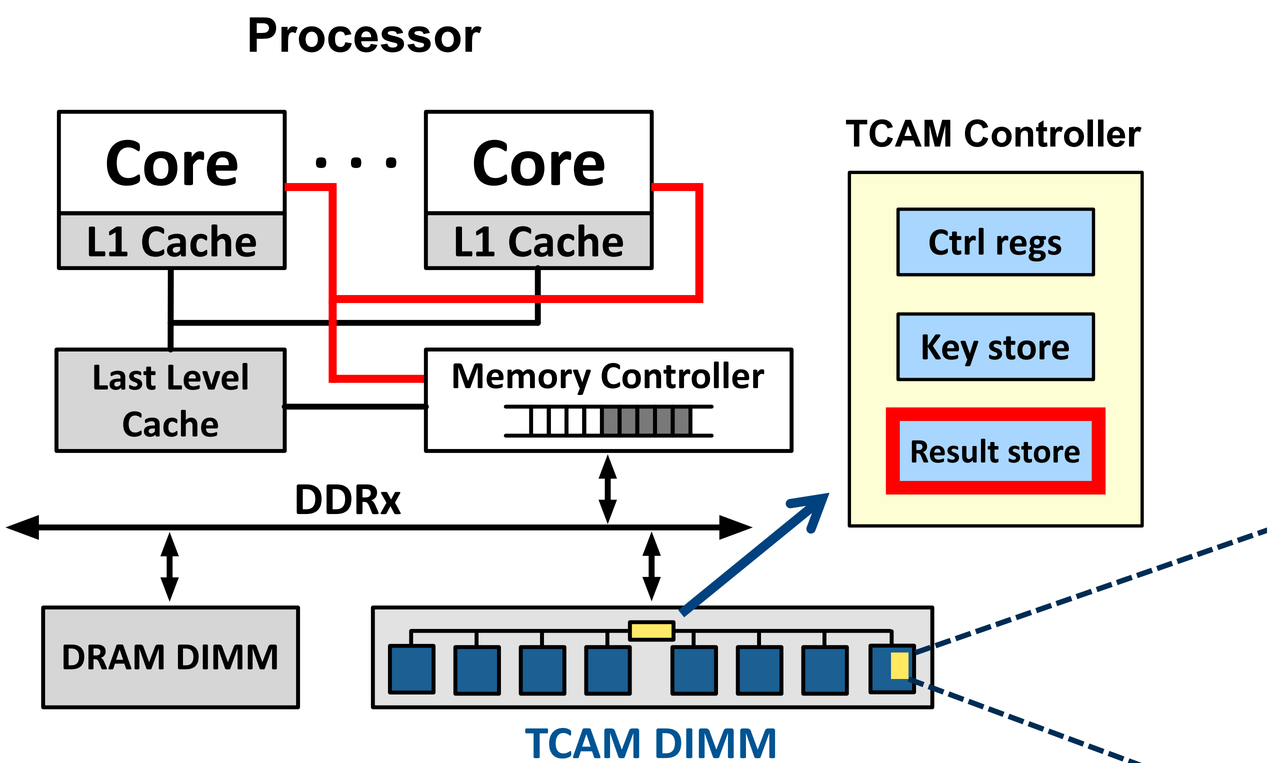
- Processor uses uncacheable loads and stores to access TCAM



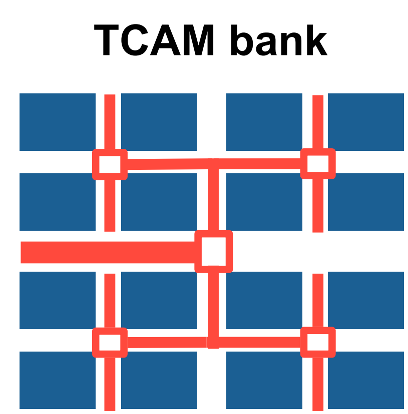
- Configure
- Update contents
- Search

Processor Interface

- Processor uses uncacheable loads and stores to access TCAM

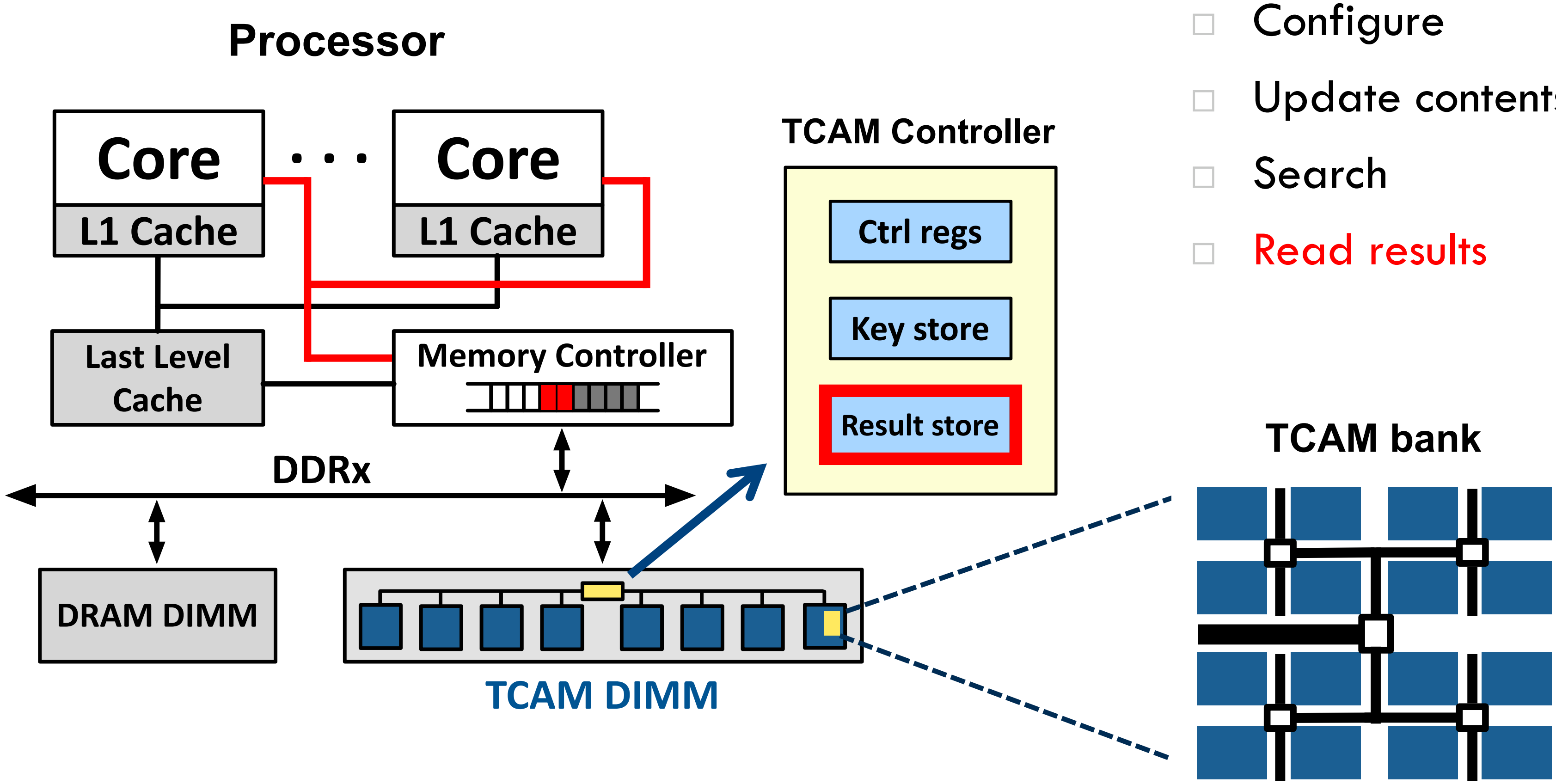


- Configure
- Update contents
- Search



Processor Interface

- Processor uses uncacheable loads and stores to access TCAM



- Configure
- Update contents
- Search
- Read results

Associative Computing Paradigm

- Broadens the use of CAMs to a more general programming framework
- Data organized by key-value pairs
  - Linked list, array, stack, queue
  - Matrix, tree, graph

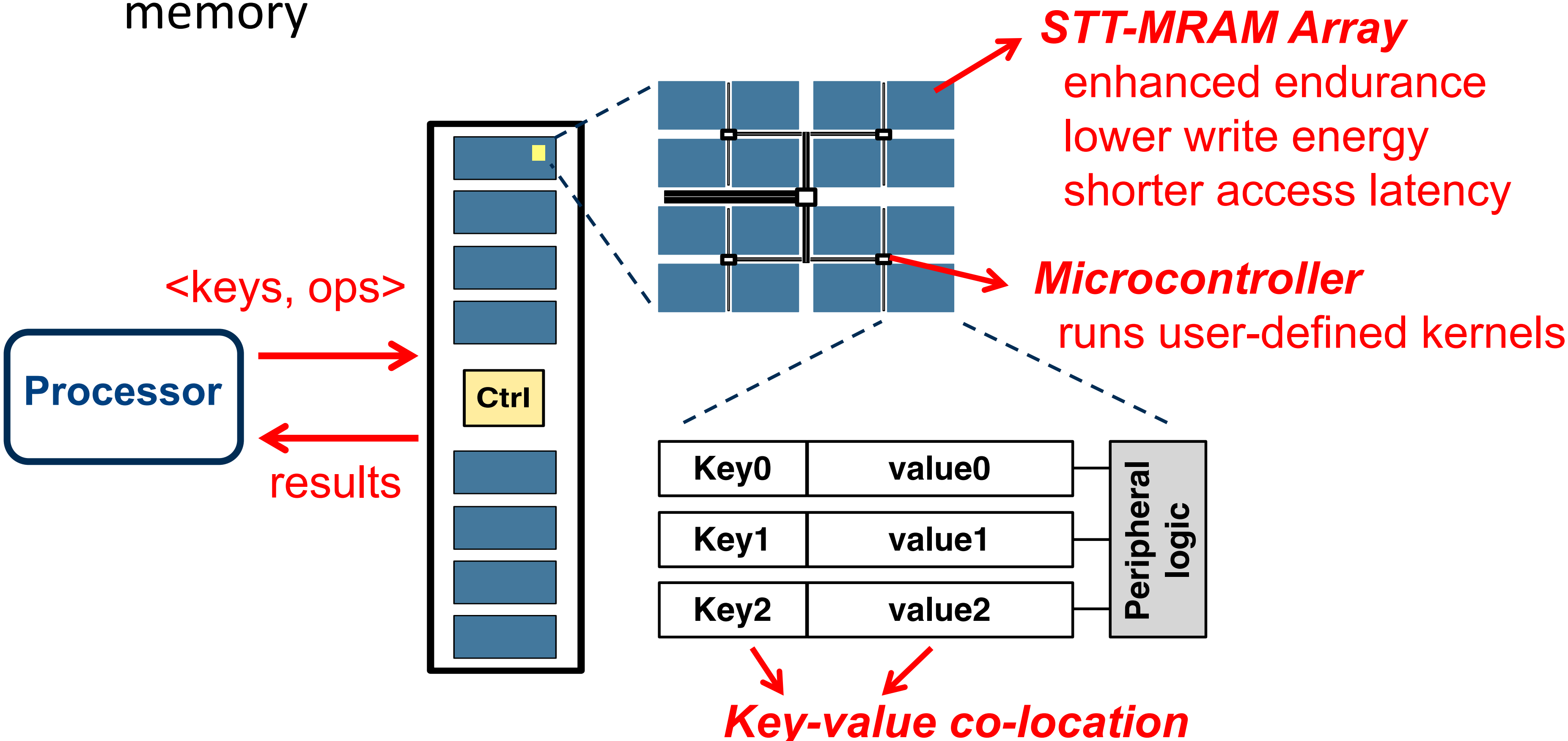
$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow$$

Key (row, col)	***Value***
(0, 0)	a
(0, 1)	b
(1, 0)	c
(1, 1)	d

L. Potter, “ASC: an associative-computing paradigm”, 1994

AC-DIMM

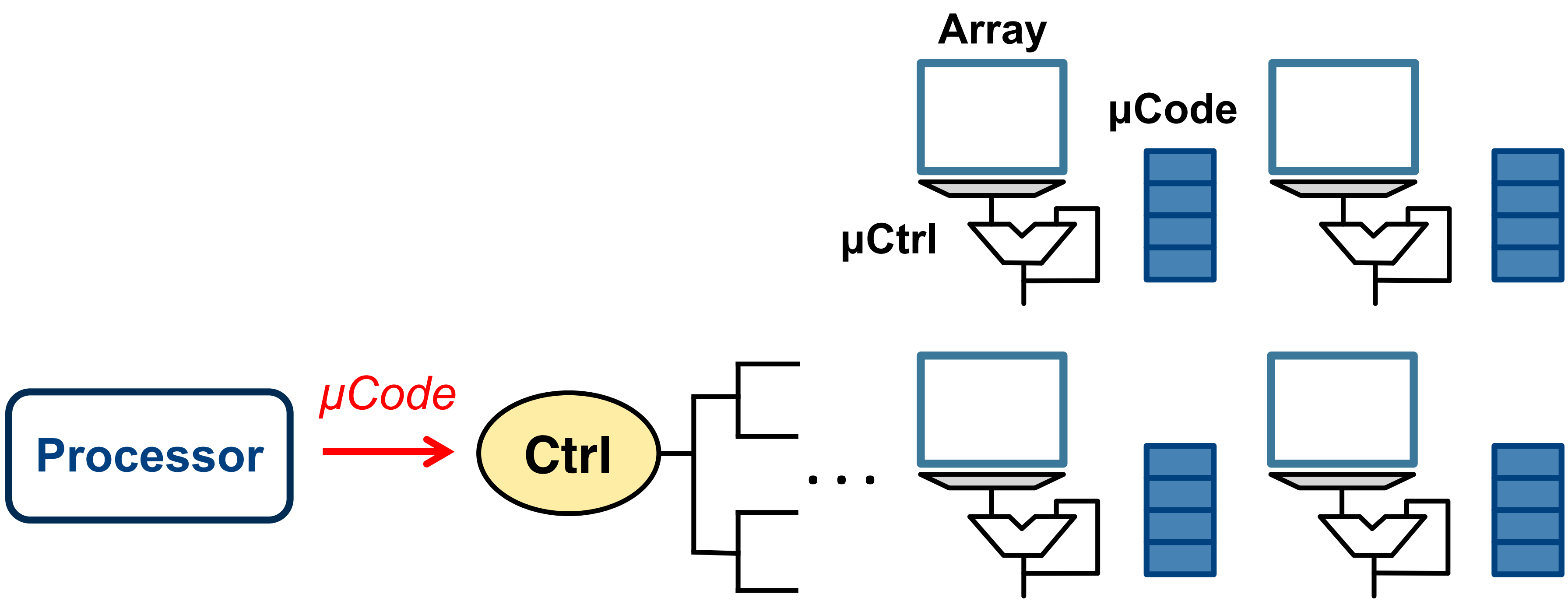
- AC-DIMM combines associative lookup and processing in memory



[ISCA'13]

Programming Model

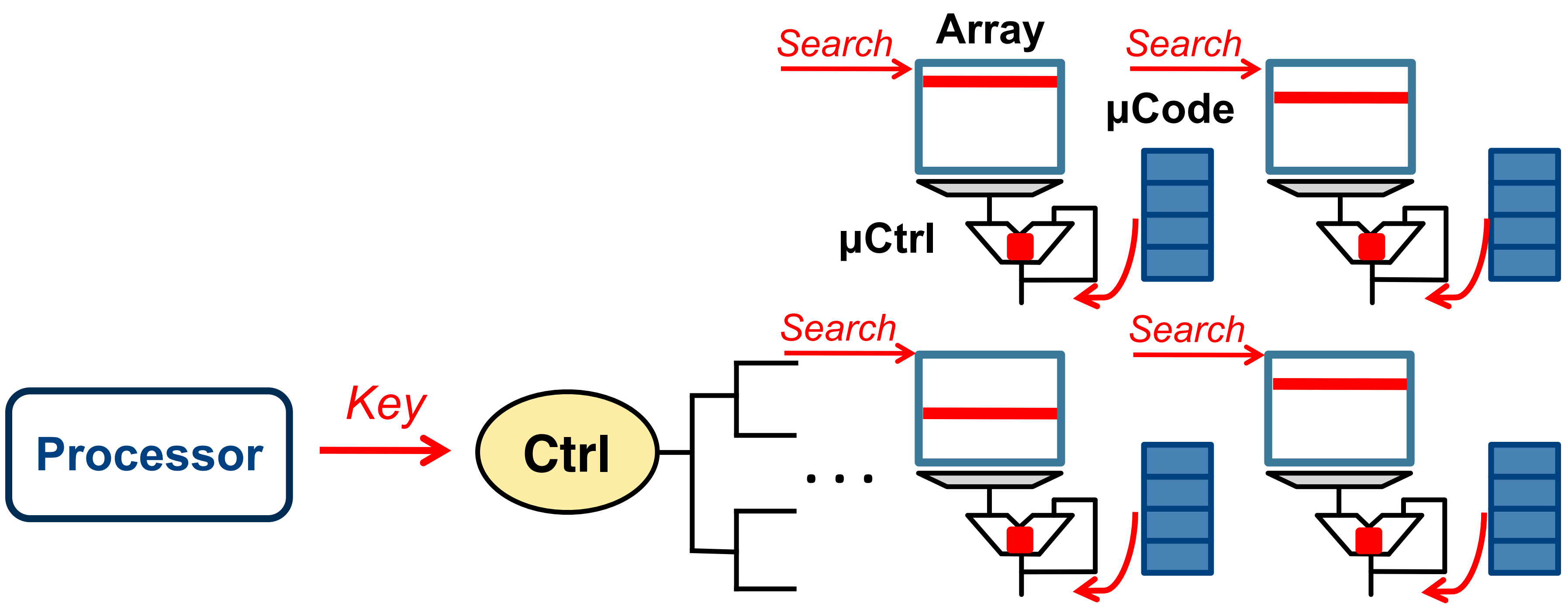
- Program accesses AC-DIMM via a user-level library



[ISCA'13]

Programming Model

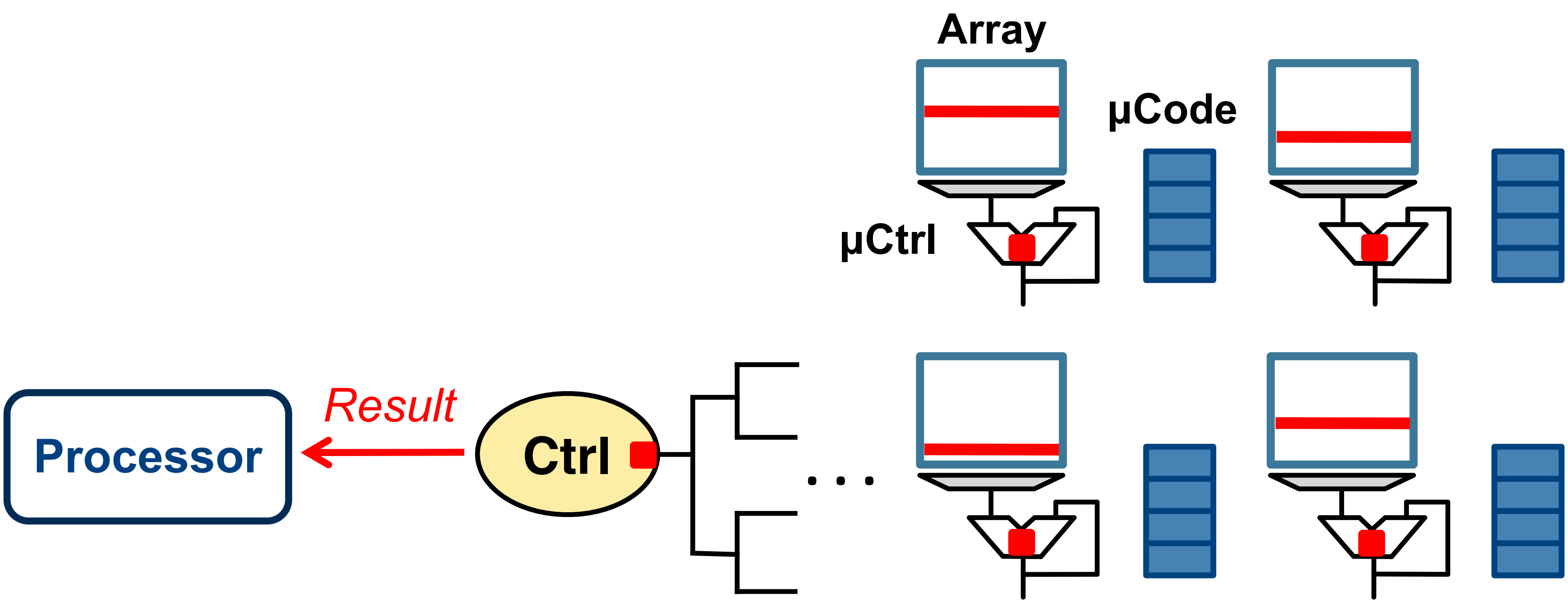
- Program accesses AC-DIMM via a user-level library



[ISCA'13]

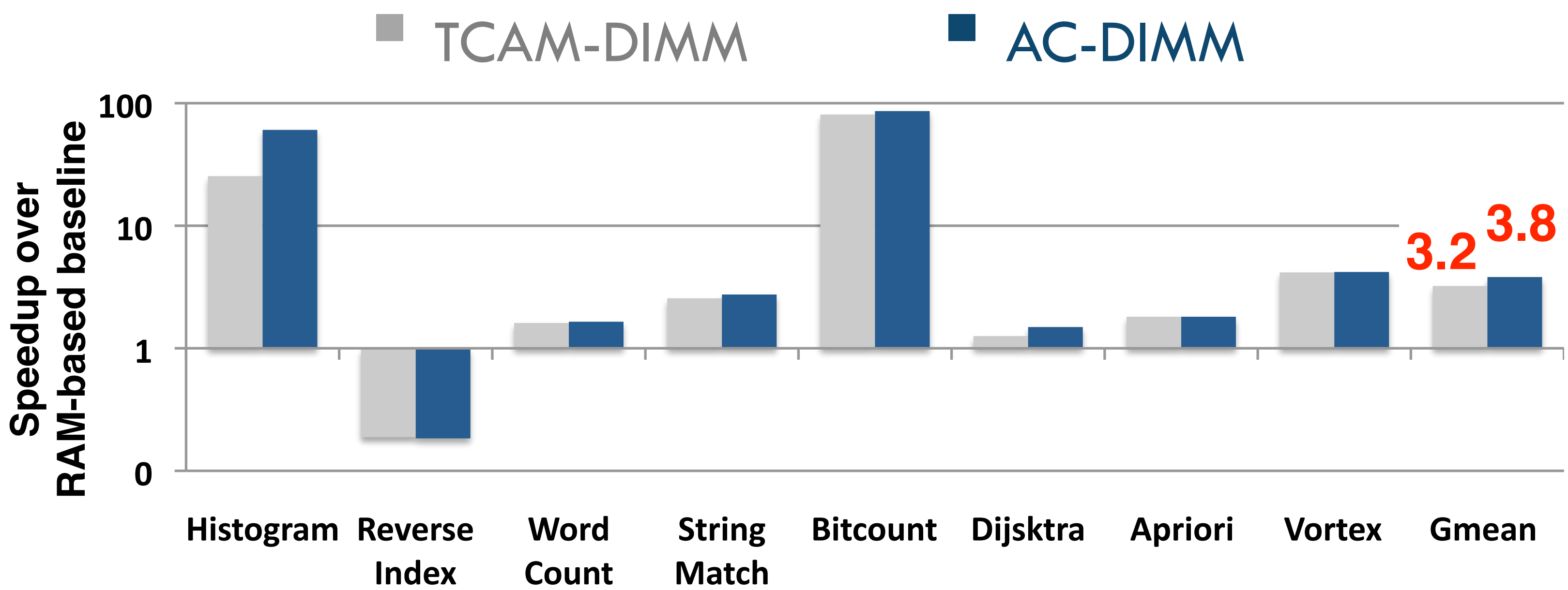
Programming Model

- Program accesses AC-DIMM via a user-level library



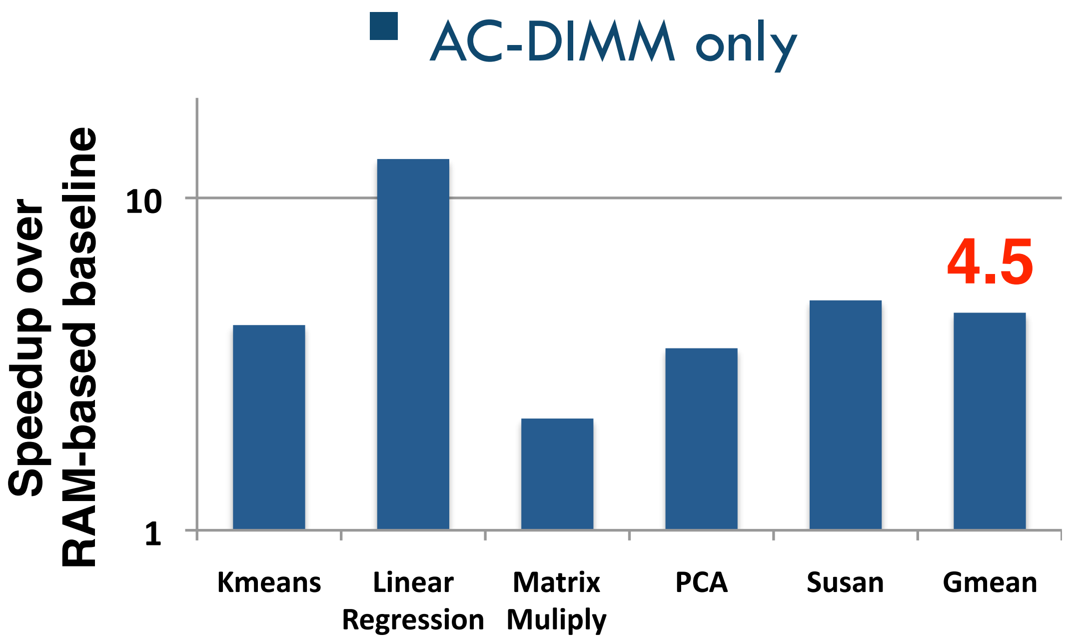
[ISCA'13]

System Performance



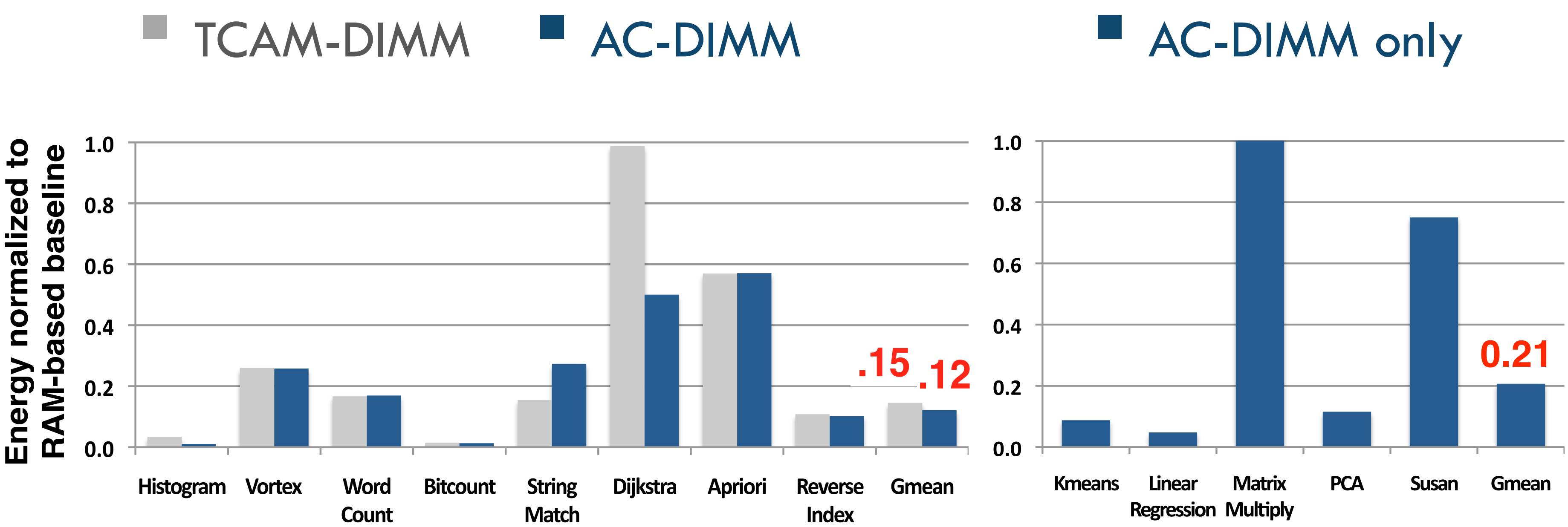
- AC-DIMM outperforms the previous TCAM-DIMM when the search key is short (<32 bits)

System Performance



- AC-DIMM caters to a broader range of applications

System Energy



- Dynamic energy saved by eliminating data movement
- Leakage energy saved by reducing execution time