

# *RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization*

Vivek Seshadri, Yoongu Kim, Chris Fallin, Donghyuk Lee,  
Rachata Ausavarungnirun, Gennady Pekhimenko, Yixin Luo,  
Onur Mutlu, Phillip B. Gibbons, Michael A. Kozuch, Todd C. Mowry

Carnegie Mellon University and Intel (Pittsburgh)

MICRO - 2013

# Outline

## Motivation

- Bottleneck
- Goal
- Observation
- Proposed Idea

## DRAM

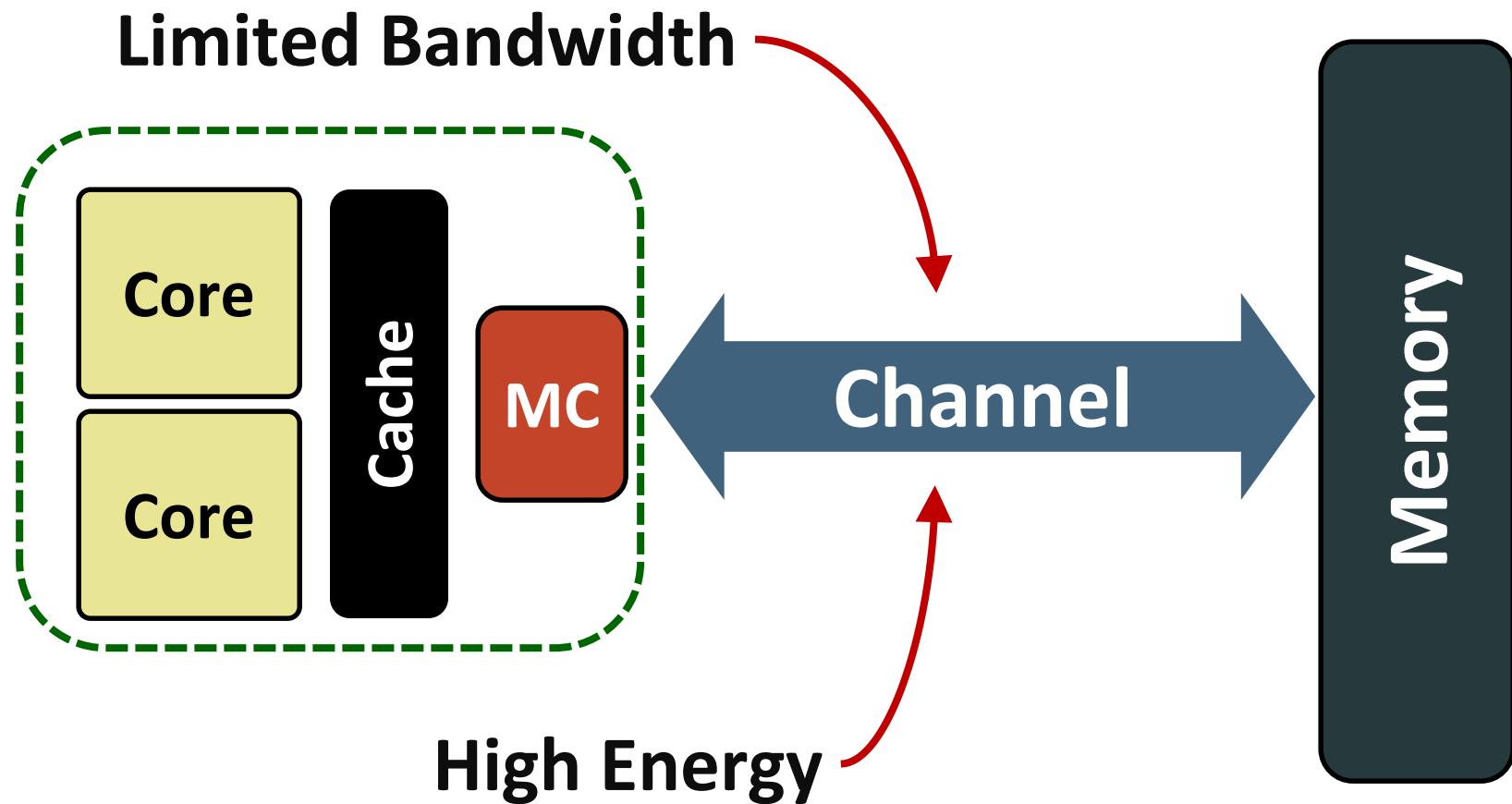
- Structure
- Operation

## RowClone

- Fast Parallel Mode
  - Limitation
- Pipelined Serial Mode

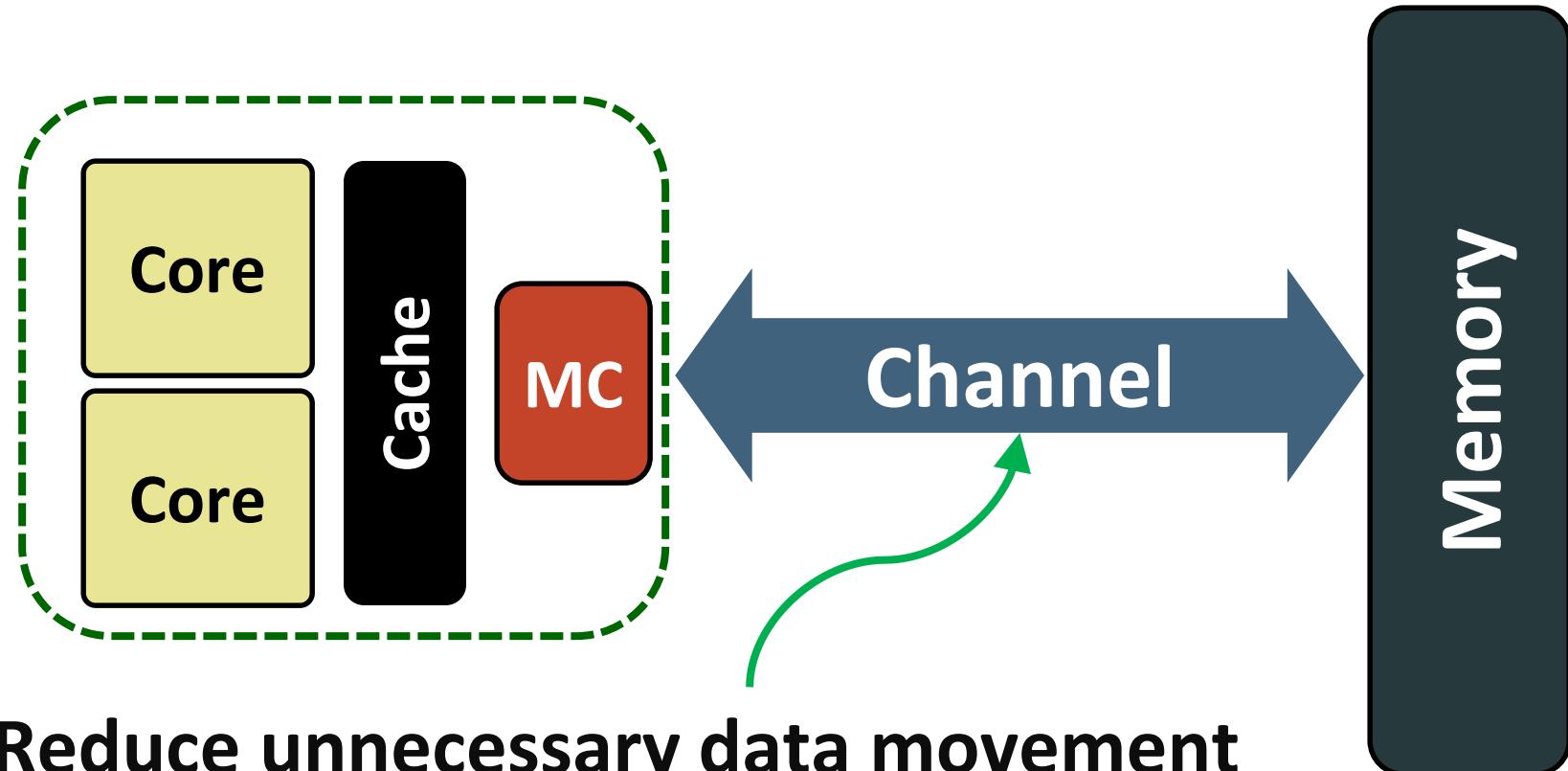
## Evaluation Results

# Bottleneck: Memory Channel



# Motivation

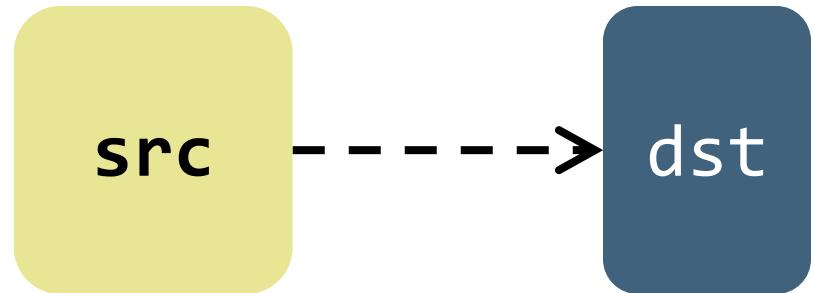
**Goal: Reduce Memory Bandwidth Demand**



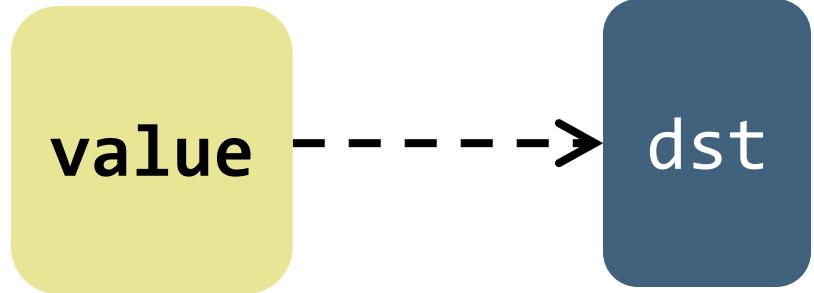
# Motivation

## Observation: Bulk Data Copy and Initialization

Bulk Data Copy

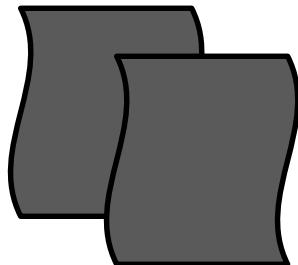


Bulk Data Initialization

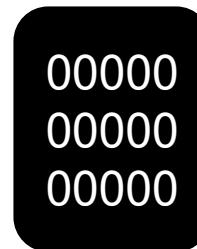


Neither requires any computation !

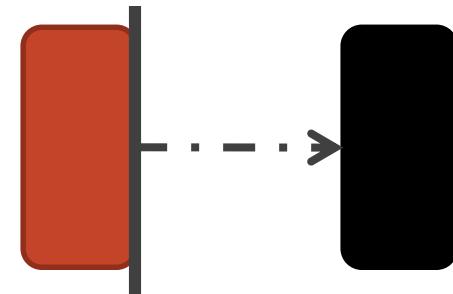
# Many Applications involve Bulk Copy and Initialization



**Forking**



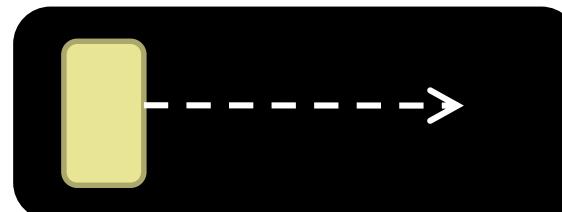
**Zero initialization**  
(e.g., security)



**Checkpointing**



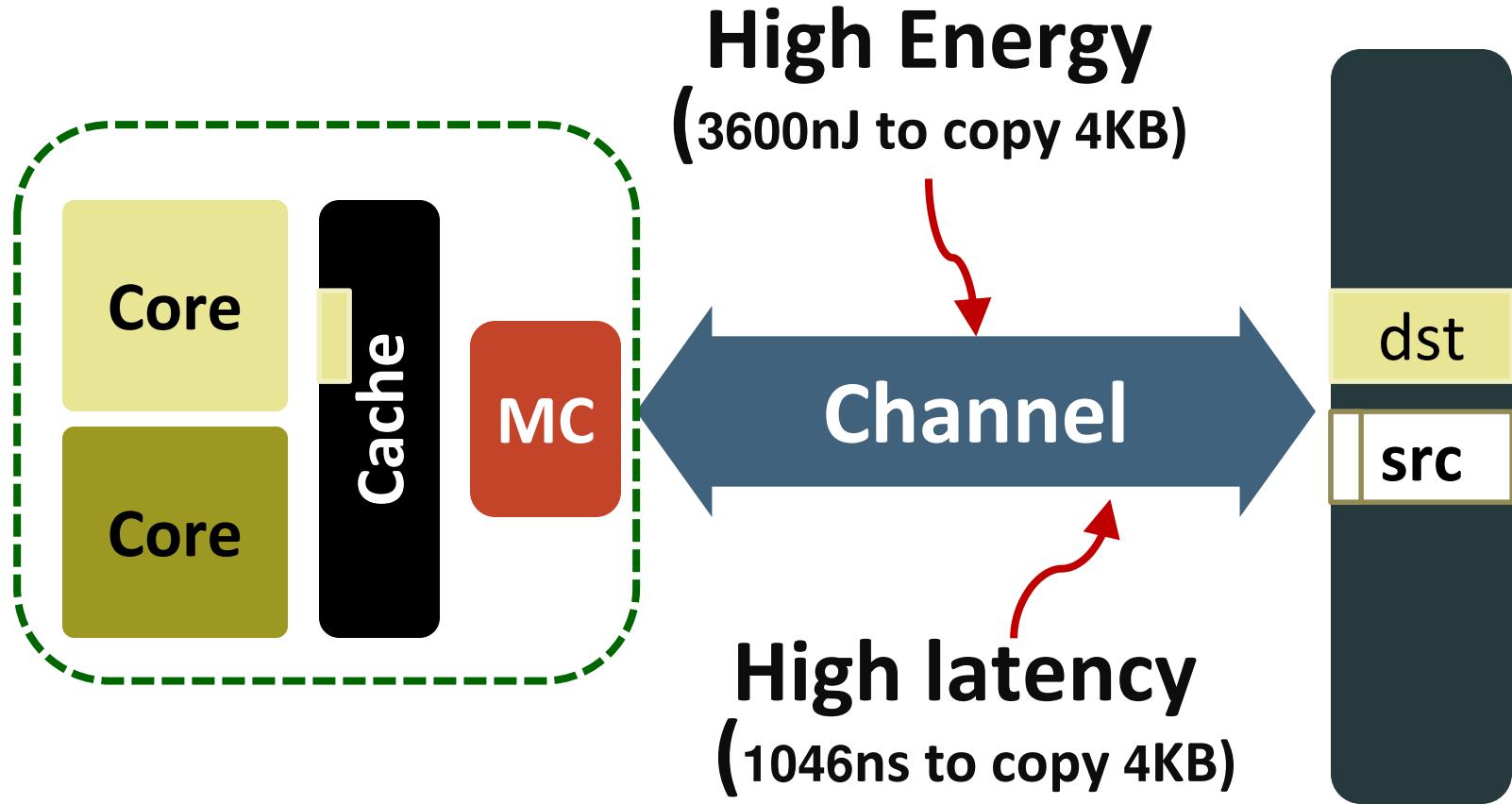
**VM Cloning**  
**Deduplication**



**Page Migration**

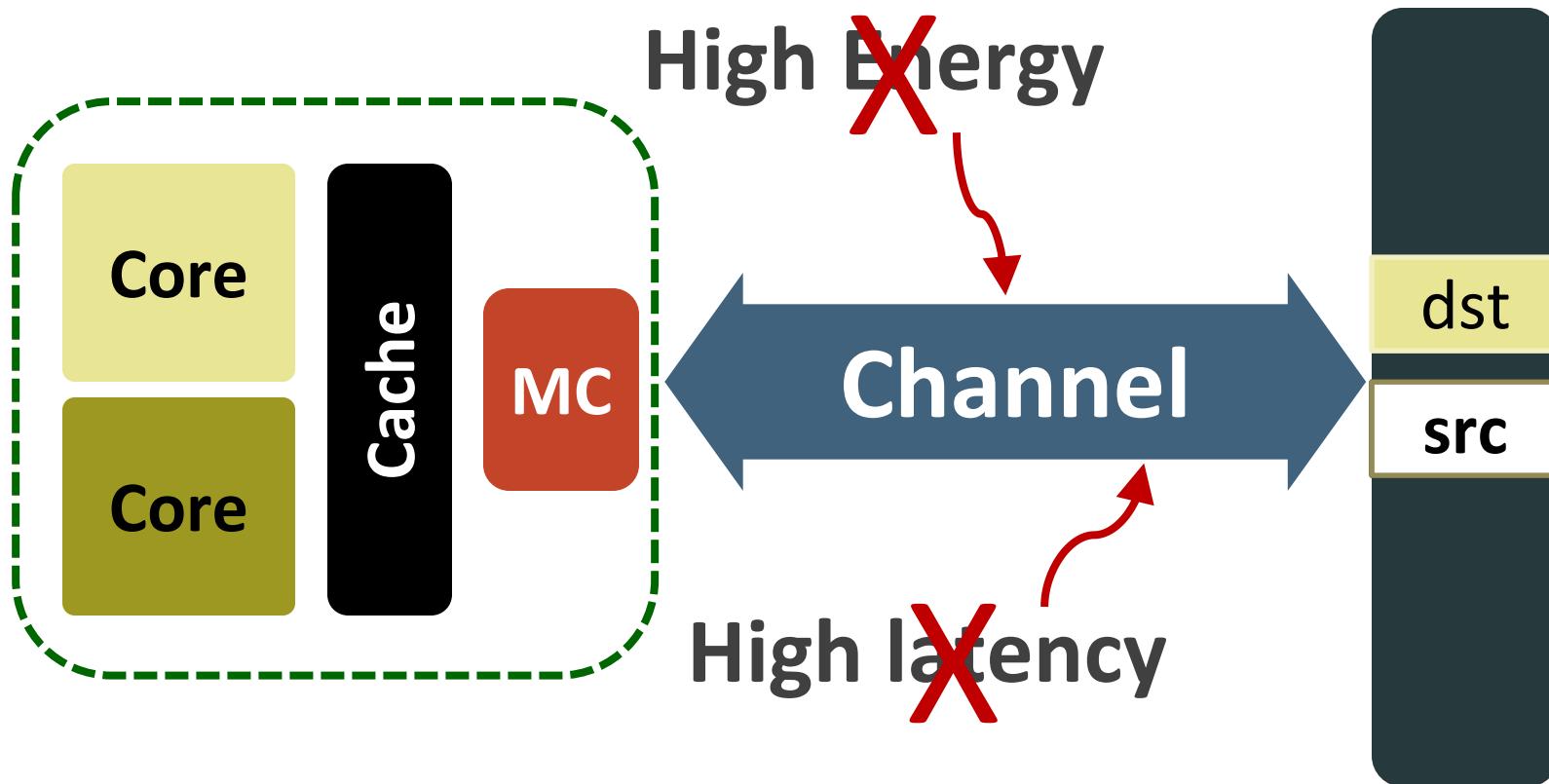
• • •  
Many more

# Weakness of Current Process

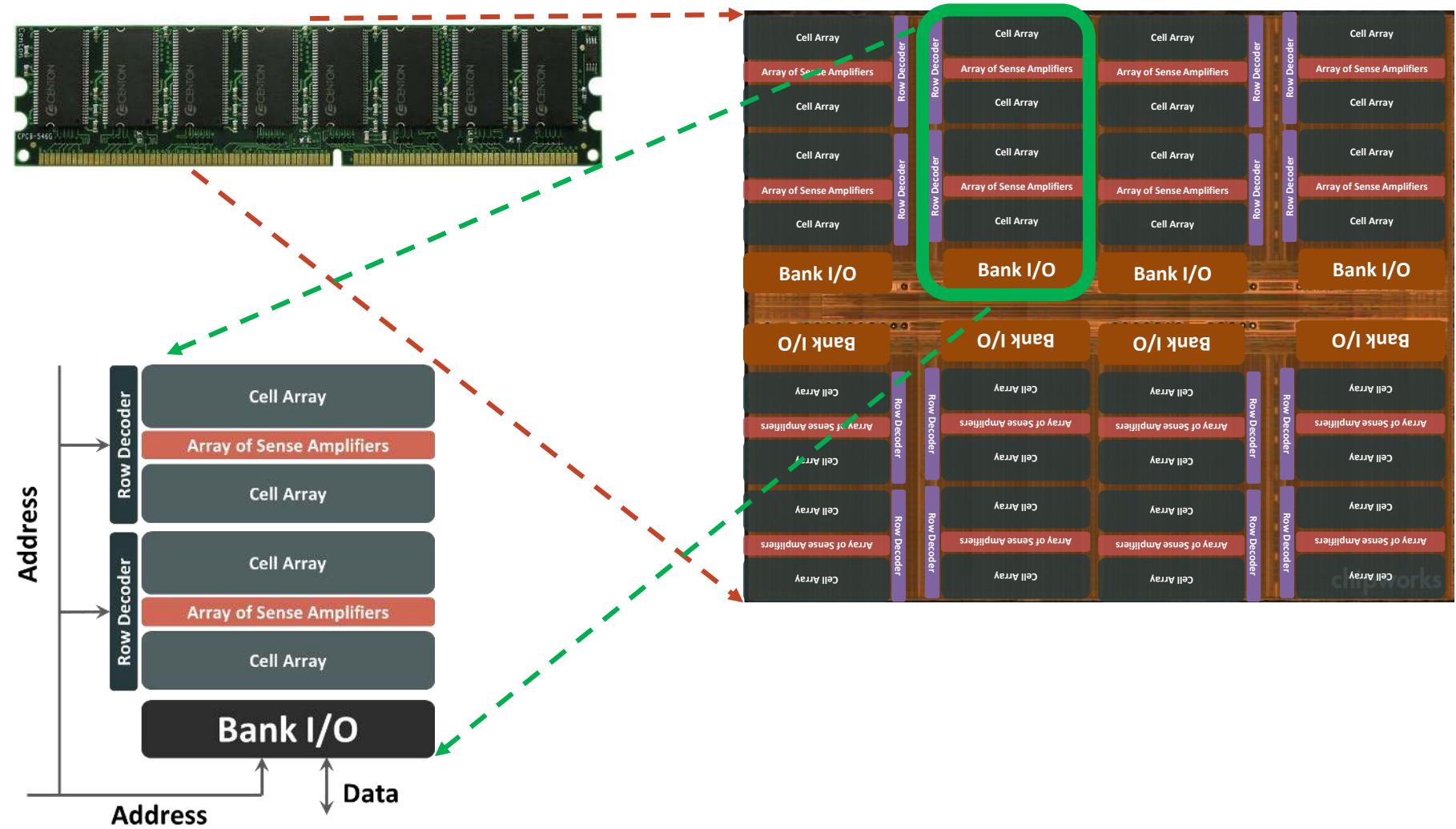


# Proposed Idea:

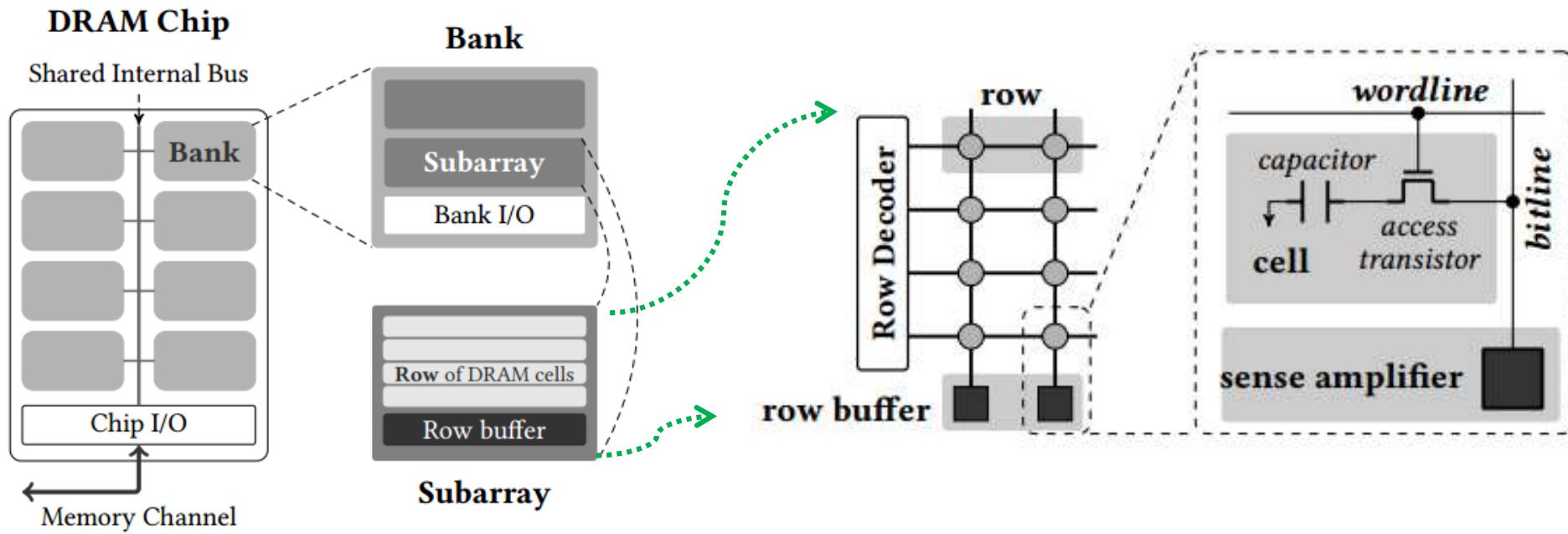
## Row Cloning → In-DRAM Copy



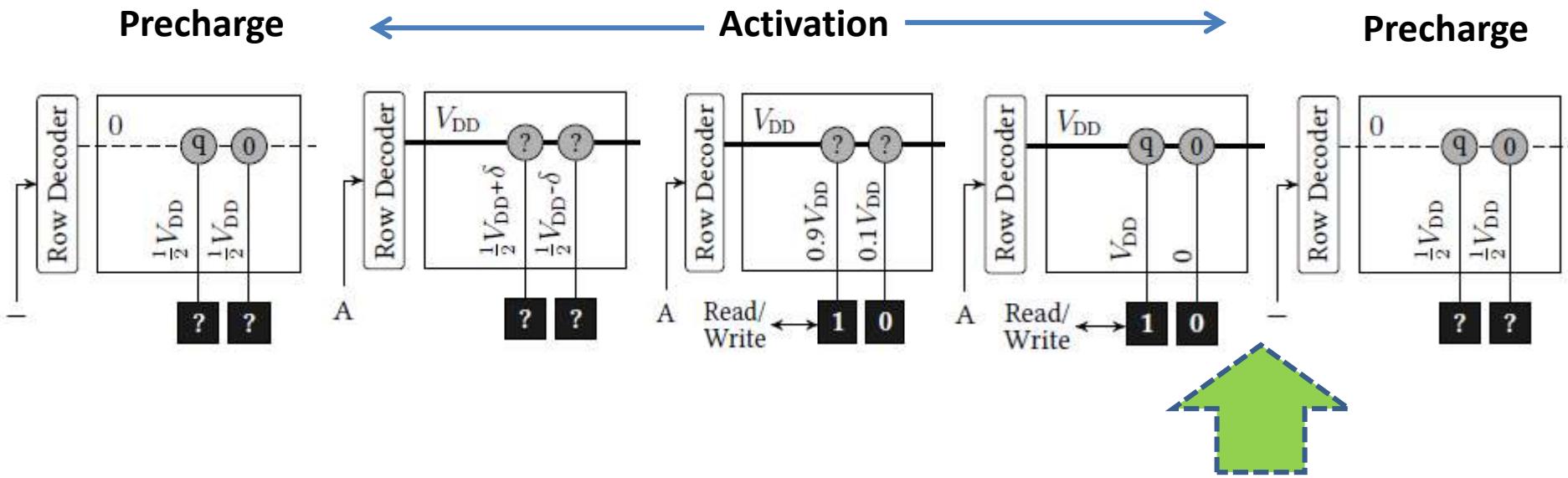
# Inside DRAM



# Digging In Little Deeper



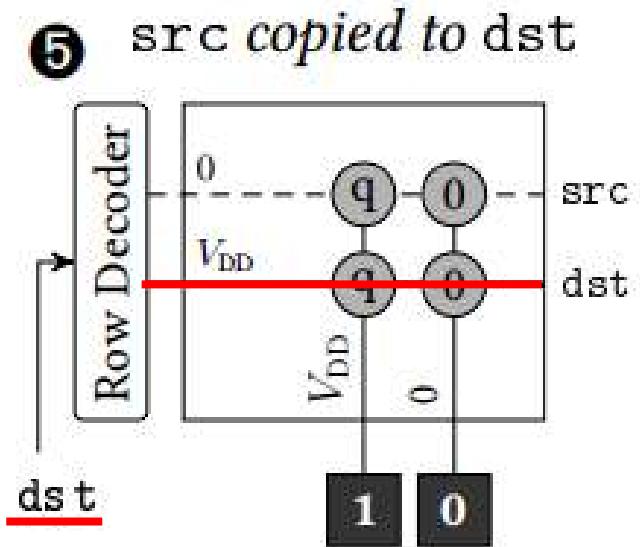
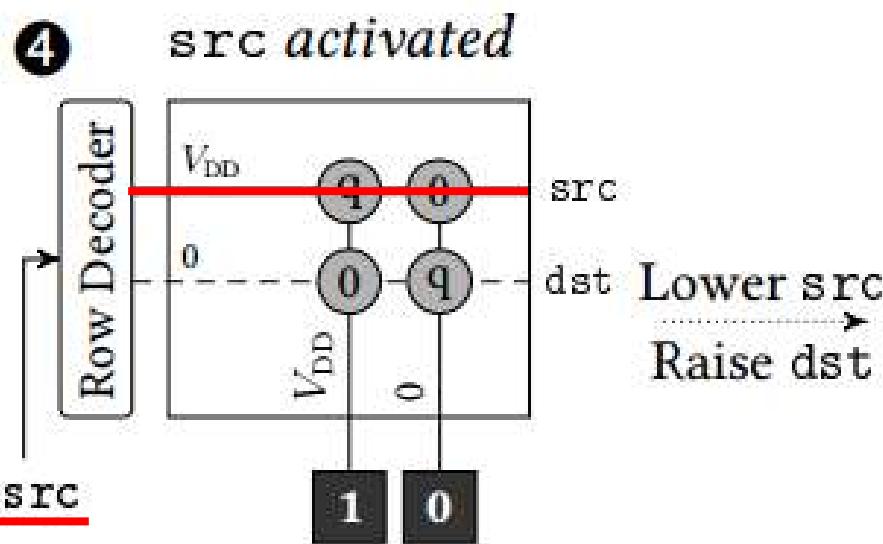
# DRAM Operation



This is where  
RowClone comes  
into the picture

# Row Clone: *Fast Parallel Mode (FPM)*

- DRAM can transfer an entire row of data from the DRAM cells to the corresponding row buffer
- FPM first copies the data from the source row to the row buffer and then copies the data from the row buffer to the destination row



## *Fast Parallel Mode - Limitations*

**It only works if the source and destination rows to be within the same sub-array**

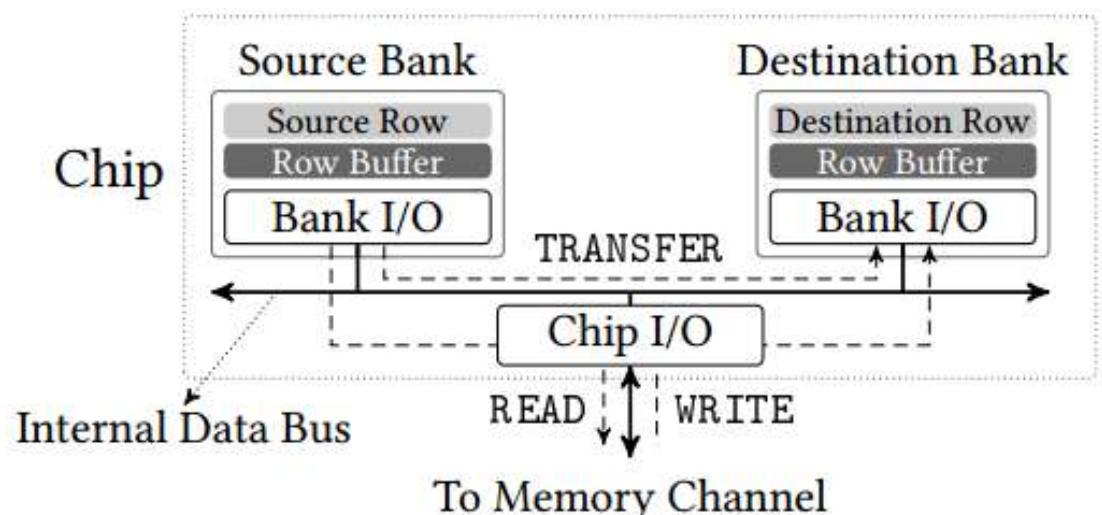
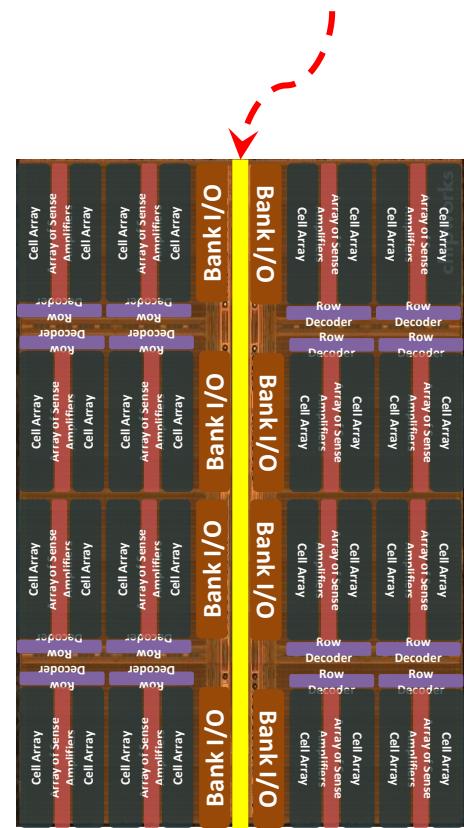
- 1. What about transfer across different banks?**
- 2. What about transfer across different sub-arrays within a bank?**

# Row Clone: *Pipelined Serial Mode (PSM)*

Shared internal bus  
across all banks

Provides an alternate mechanism to copy  
data across banks

PSM allows to transfer data without  
crossing the Chip I/O interface



# Row Clone: *Pipelined Serial Mode (PSM)*

Inter-Bank Transfer:

- 1) Activates both source and destination rows in the corresponding banks
- 2) Puts the source bank in Read mode and destination bank in Write mode to transfer data

Intra-Bank Transfer:

- 1) Use PSM to transfer data from source to a temporary row in a different bank
- 2) Again use PSM to transfer data from temporary location to the destination

# Miscellaneous

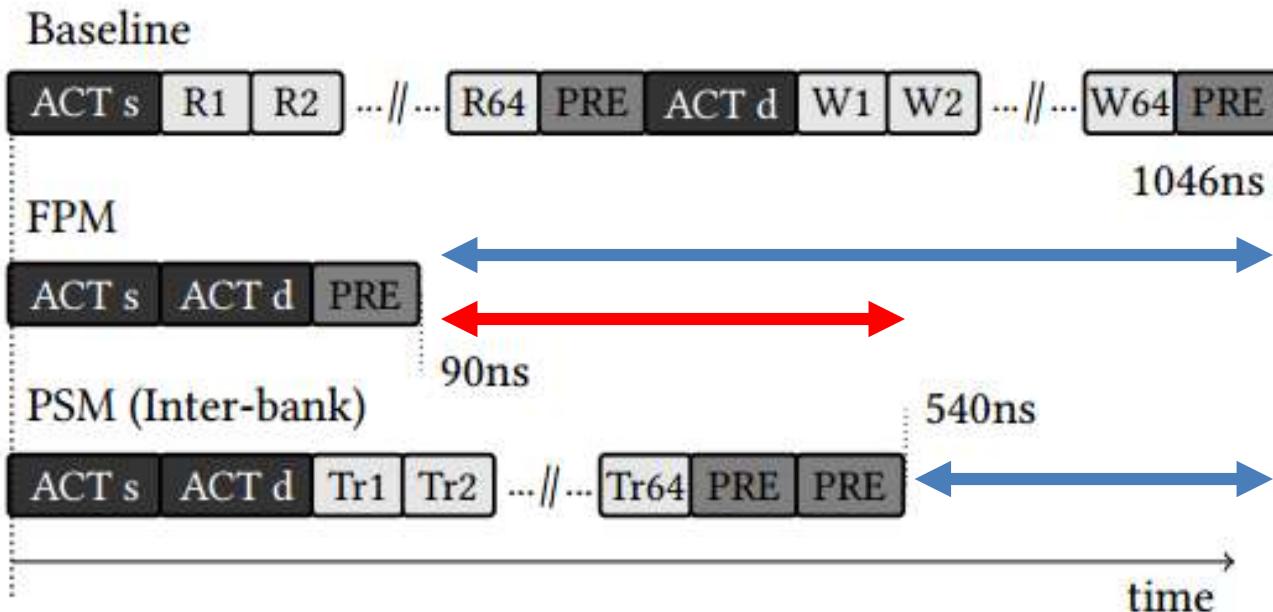
## ISA support:

Instruction	Operands	Semantics
memcpy	<i>src, dst, size</i>	Copy <i>size</i> bytes from <i>src</i> to <i>dst</i>
meminit	<i>dst, size, val</i>	Set <i>size</i> bytes to <i>val</i> at <i>dst</i>

## Row Clone - ZI:

- RowClone-Zero-Insert (RowClone-ZI) inserts a zero cache line into the processor cache corresponding to the page that is zeroed out.
- By doing this, RowClone-ZI avoids the cache misses during both zeroing operation and when the application accesses the cache lines of the zeroed page.

# Raw Latency Improvement

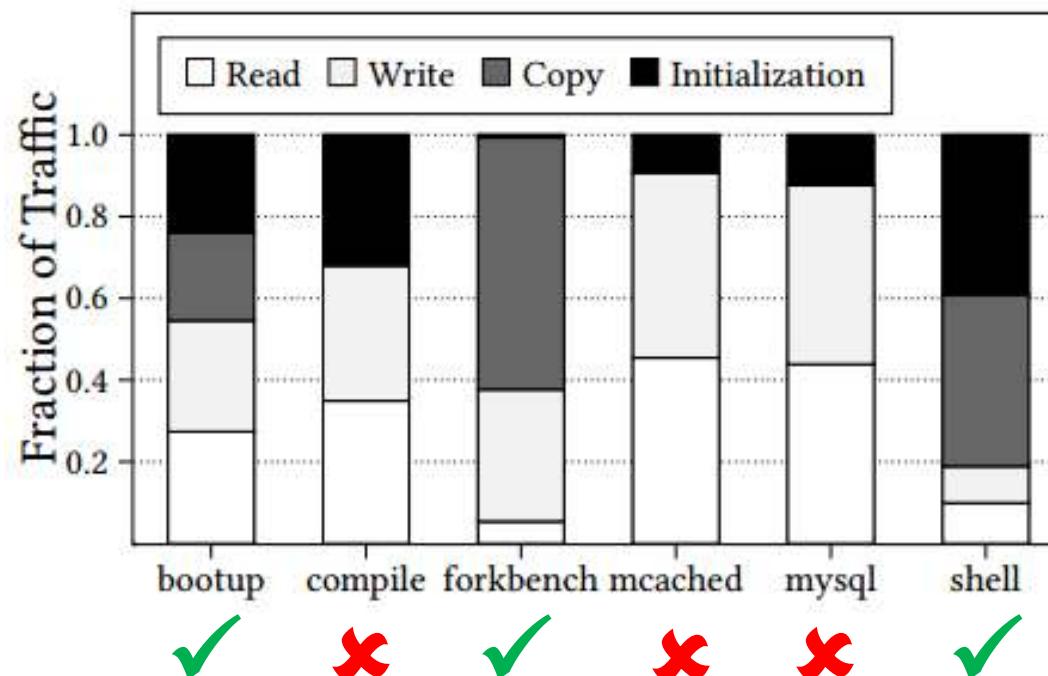


ACT s – ACTIVATE source, ACT d – ACTIVATE destination  
R – READ, W – WRITE, Tr – TRANSFER, PRE – PRECHARGE

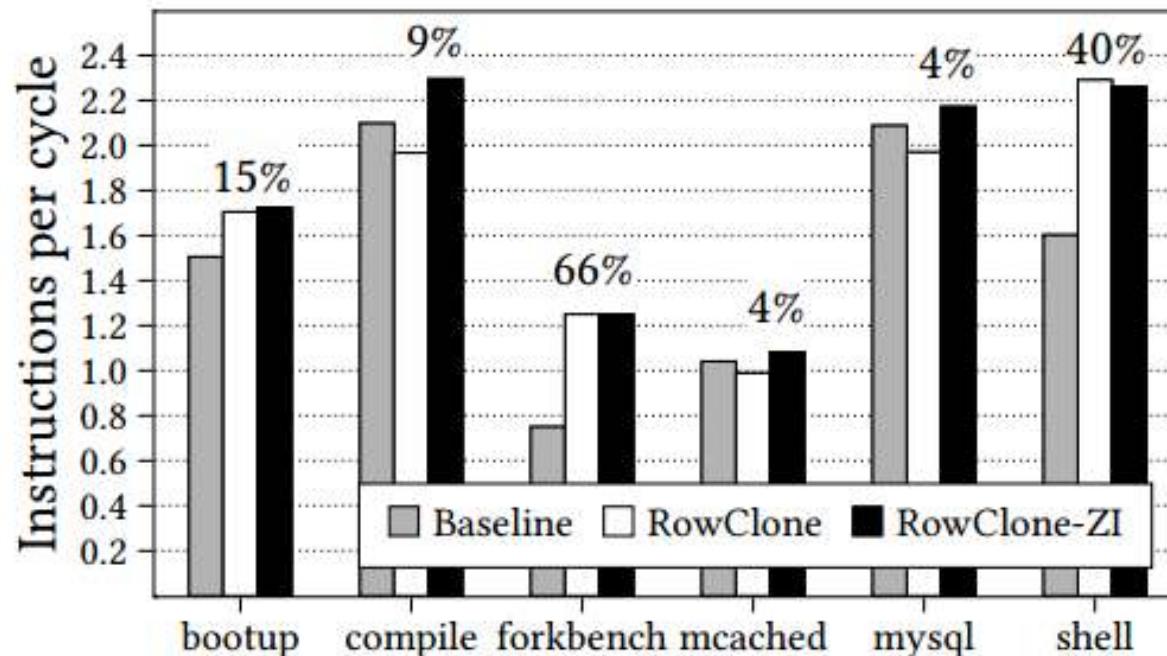
# Raw Energy Improvement

	<b>Mechanism</b>	<b>Absolute</b>		<b>Reduction</b>	
		Latency (ns)	Memory Energy ( $\mu$ J)	Latency	Memory Energy
Copy	Baseline	1046	3.6	1.00x	1.0x
	FPM	90	0.04	<b>11.62x</b>	<b>74.4x</b>
	Inter-Bank - PSM	540	1.1	1.93x	3.2x
	Intra-Bank - PSM	1050	2.5	0.99x	1.5x
Zero	Baseline	546	2.0	1.00x	1.0x
	FPM	90	0.05	<b>6.06x</b>	<b>41.5x</b>

# Traffic Distribution – Copy and Initialization



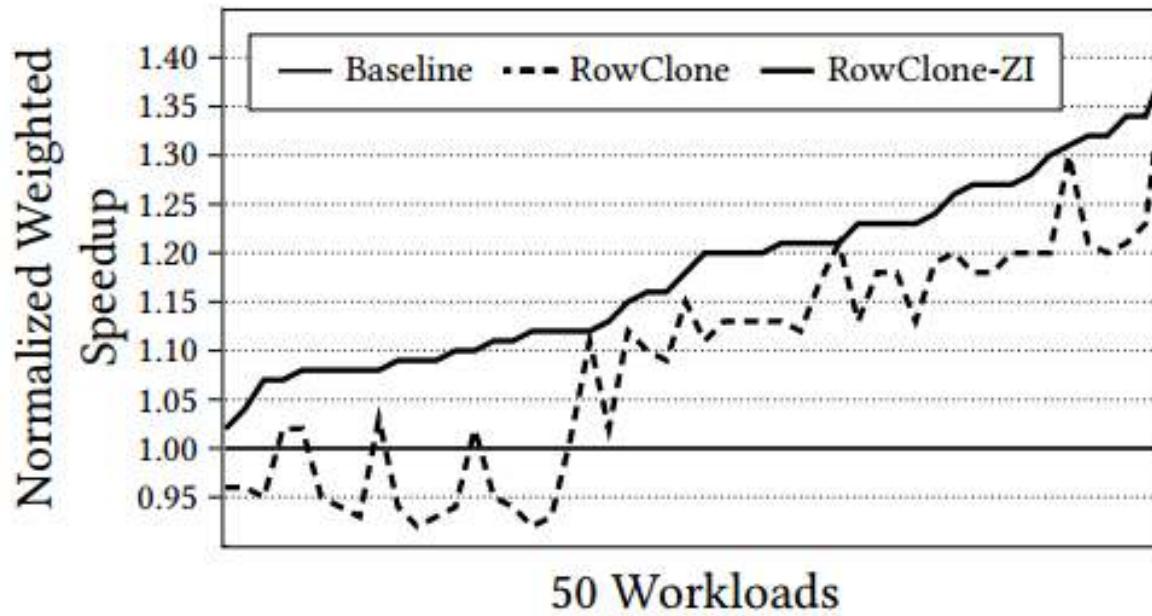
# Performance Comparison – Single-core Environment



# Energy and Bandwidth Improvement

Application	Energy Reduction		Bandwidth Reduction		
	RowClone	+ZI	RowClone	+ZI	
<i>bootup</i>	39%	40%	49%	52%	✓
<i>compile</i>	-2%	32%	2%	47%	✗
<i>forkbench</i>	69%	69%	60%	60%	✓
<i>mcached</i>	0%	15%	0%	16%	✗
<i>mysql</i>	-1%	17%	0%	21%	✗
<i>shell</i>	68%	67%	81%	81%	✓

# Multi-core Performance



- RowClone improves performance for most workloads
- RowClone-ZI improves workload for all workloads

# Thank you