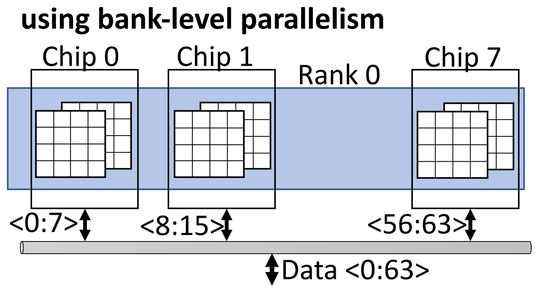
PALP: Enabling and Exploiting Partition-Level Parallelism in Phase Change Memories

Shihao Song, Anup Das, Onur Mutlu and Nagarajan Kandasamy

Drexel University ETH zürich

Background: Bank Conflicts

- PCM devices can serve multiple requests in parallel using bank-level parallelism



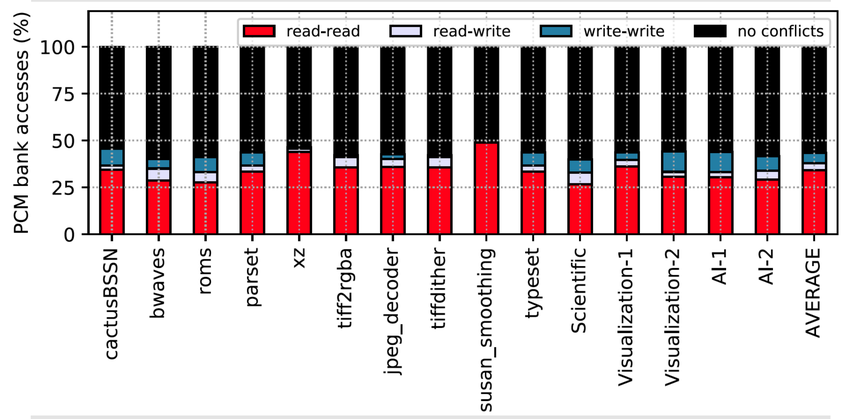
- Requests to the same bank have to be served serially, known as bank conflict

Bank conflicts occur due temporal and spatial access locality in a workload that lead to repeated access to multiple memory rows that map to the same bank

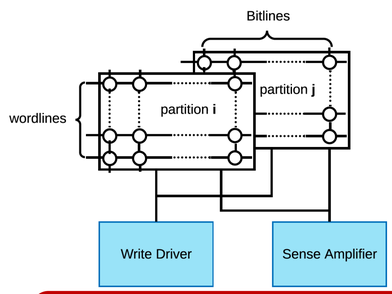
- Memory bank conflicts reduce system performance by lowering bank utilization, causing CPU cores to stall

Key Observations

1st key observation: On average, 43% of PCM requests in SPEC 2017 workloads generate bank conflicts



2nd key observation: A PCM bank is implemented as a collection of partitions that operate mostly independently; sharing the sense amplifiers (to read) and the write drivers (to write)



- The peripheral structures in PCM banks allow to read and program 128 PCM cells in parallel

Goals

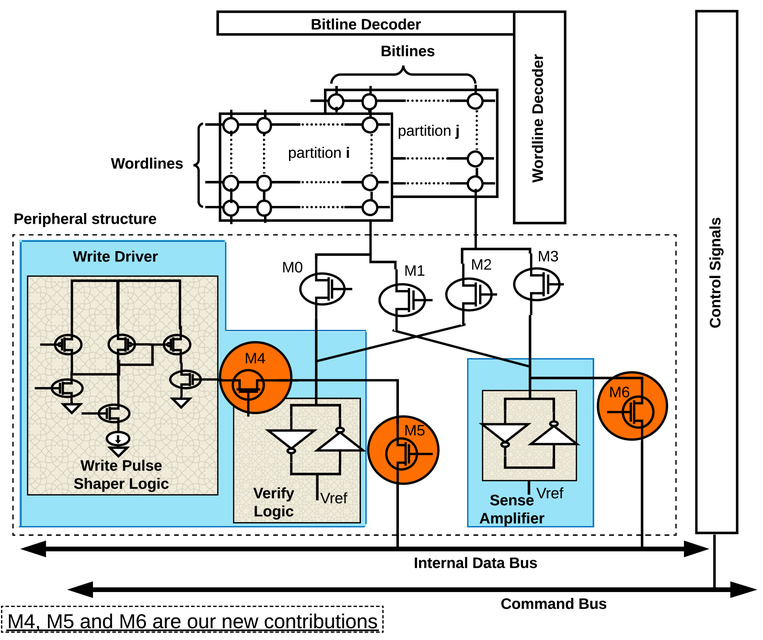
1. Analyze parallelism in PCM bank's partition through detailed circuit analysis
2. Analyze the potential performance impact of enabling parallelism in a bank's partitions
3. Resolve read-read and read-write PCM bank conflicts to improve PCM performance

Our Contributions

1. New mechanism to resolve read-write PCM bank conflicts using a new PCM command called READ-WITH-WRITE (RWW)
2. Simple circuit modification to resolve read-read PCM bank conflicts using a new PCM command called READ-WITH-READ (RWR)
3. New memory access scheduling to prioritize PCM requests that exploit a PCM bank's partition-level parallelism, over other requests, including the long outstanding ones

Enabling and Exploiting Partition-Level Parallelism in PCM

Decoupled Write Driver



M4, M5 and M6 are our new contributions

Operating Modes	Write driver circuit: Pulse shaper logic	Write driver circuit: Verify logic	Sense Amplifier	Operations Enabled
write	√	√	×	one write request
decoupled	×	√	√	two read requests

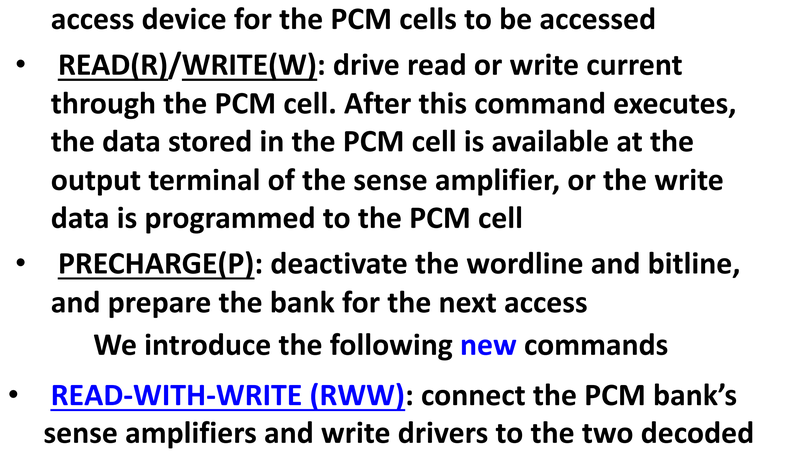
PCM Commands

- ACTIVATE(A): activate the wordline and enable the access device for the PCM cells to be accessed
- READ(R)/WRITE(W): drive read or write current through the PCM cell. After this command executes, the data stored in the PCM cell is available at the output terminal of the sense amplifier, or the write data is programmed to the PCM cell
- PRECHARGE(P): deactivate the wordline and bitline, and prepare the bank for the next access

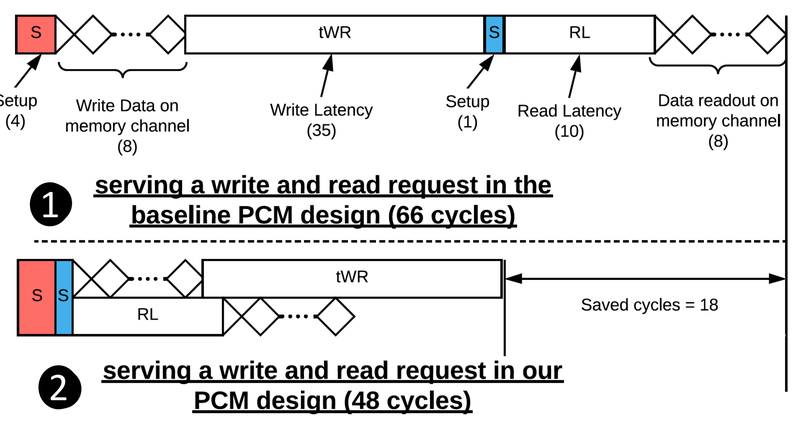
We introduce the following new commands

- READ-WITH-WRITE (RWW): connect the PCM bank's sense amplifiers and write drivers to the two decoded partitions
- READ-WITH-READ (RWR): connect the sense amplifiers and verify logic of the write drivers to the two decoded partitions
- DECOUPLE (D): set M4 = OFF
- TRANSFER (T): sets M5 = ON and M6 = OFF

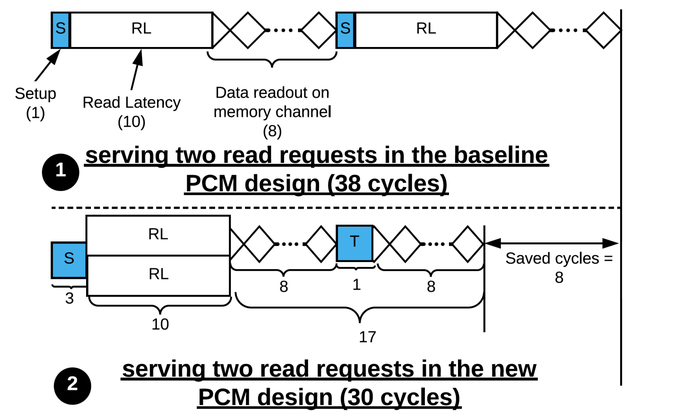
PCM Access Scheduling



Performance Improvement vs. Baseline



Performance improvement 27%



Performance improvement 21%

Evaluation

Evaluation Methodology

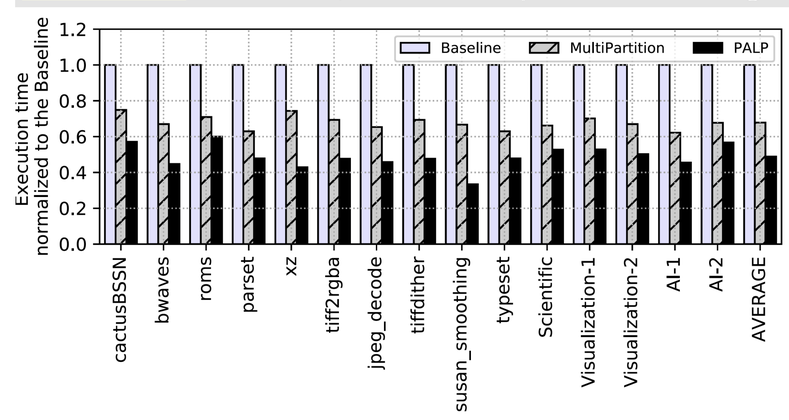
1. Gem5 frontend to simulate ARMv8-A (aarch64) with 8 cores
2. Hybrid DRAM-PCM memory system
3. In-house cycle-level PCM simulator for 8GB, 16GB, and 32GB PCM with DDR4 interface

MiBench workloads [20]
tiff2rgba, jpeg_decode, tiffdither, susan_smoothing, typeset

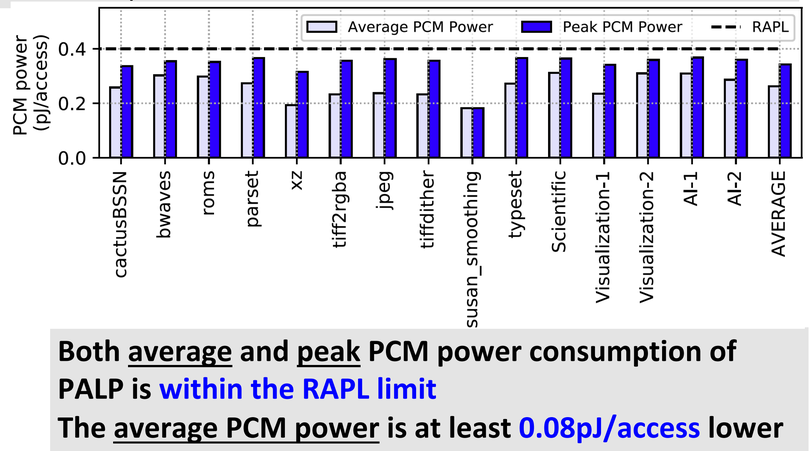
SPEC CPU2017 workloads [8]
cactusBSSN, bwaves, roms, parset, xz

Mixed (parallel) applications
AI-1 (4 copies each of deepsjeng, leela), **AI-2** (4 copies each of mcf, exchange2), **Visualization-1** (4 copies each of povray, blender), **Visualization-2** (4 copies each of povray, imagick), **Scientific** (4 copies each of cactusBSSN, bwaves)

Baseline: Arjomand et al. "*Boosting access parallelism to PCM-based main memory*" in ISCA 2016
MultiPartition: Zhou et al., "An efficient parallel scheduling scheme on multi-partition PCM architecture" in ISLPED 2016



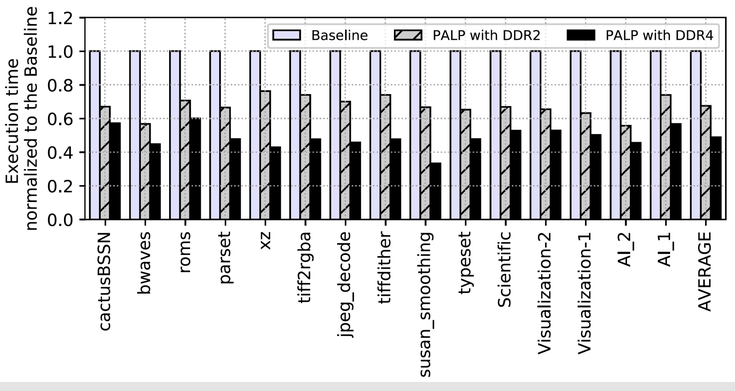
On average, execution time reduces by 51% vs. Baseline and 28% vs. MultiPartition



Both average and peak PCM power consumption of PALP is within the RAPL limit
The average PCM power is at least 0.08pJ/access lower than the RAPL limit while the peak PCM power is at least 0.03pJ/access lower than the RAPL limit

Design	Technology Type	Vdd (V)	Critical Path Delay (ps)	Power (pJ/access)
Baseline [37]	Double-gate	0.9	1159.2	0.311
Proposed PALP	Double-gate	0.9	1453.2	0.364

On average, critical path delay increases by 25.3%
On average, power consumption increases by 17%



On average, execution time reduces by 33% with DDR2 and 51% with DDR4

Open source tool
https://github.com/drexel-DISCO/PALP