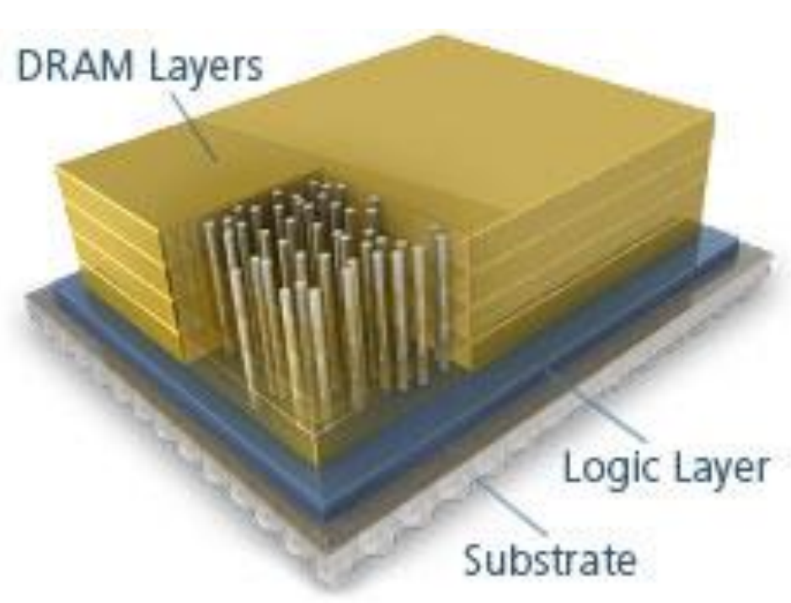
Microscope on Memory: MPSoC-enabled Computer Memory System Assessments

Abhishek Kumar Jain, Scott Lloyd, Maya Gokhale (Center for Applied Scientific Computing, LLNL)

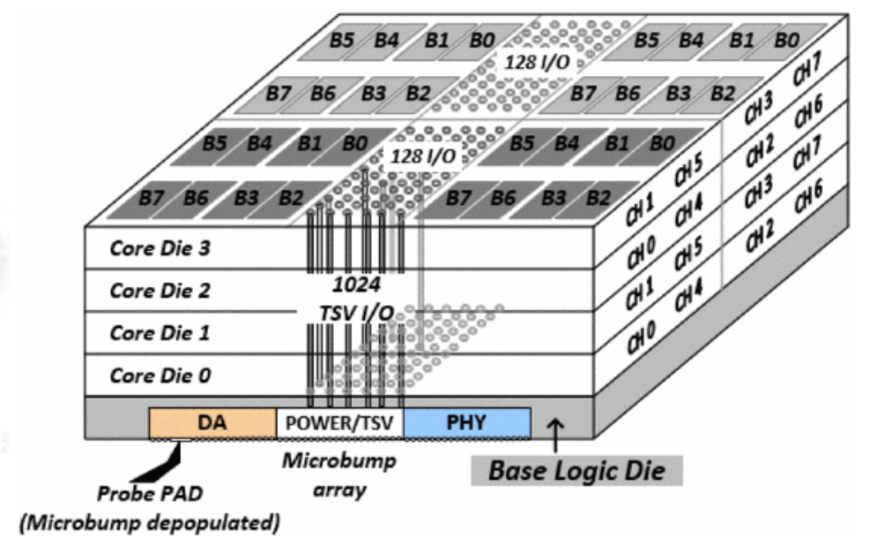
Introduction

- Emerging memories display a wide range of bandwidths, latencies, and capacities
- Potential for logic and compute functions co-located with the memory
- Challenging for the computer architect to navigate the design space of potential memory configurations
- Challenging for the application developer to assess performance implications
- Trace-driven simulation using architecture simulators (such as gem5) – very slow
- Emulation of complete system on FPGAs – Fast but labor intensive
- Our approach: Use embedded CPU cores and cache hierarchy in MPSoC as components for developing Logic in Memory Emulator (LiME)

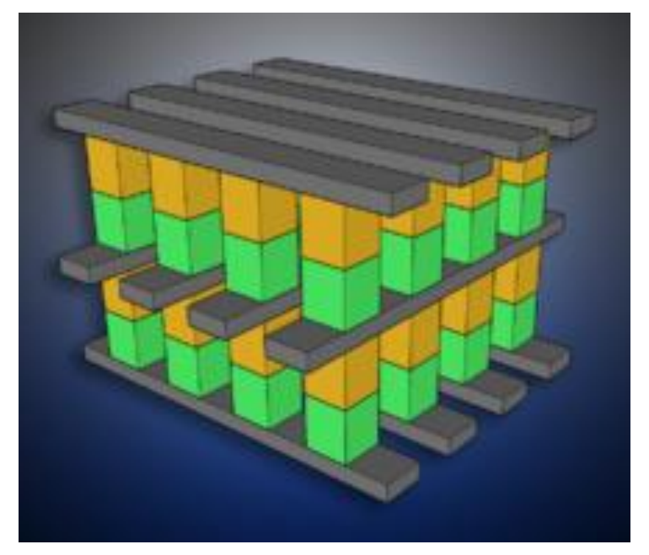
HMC

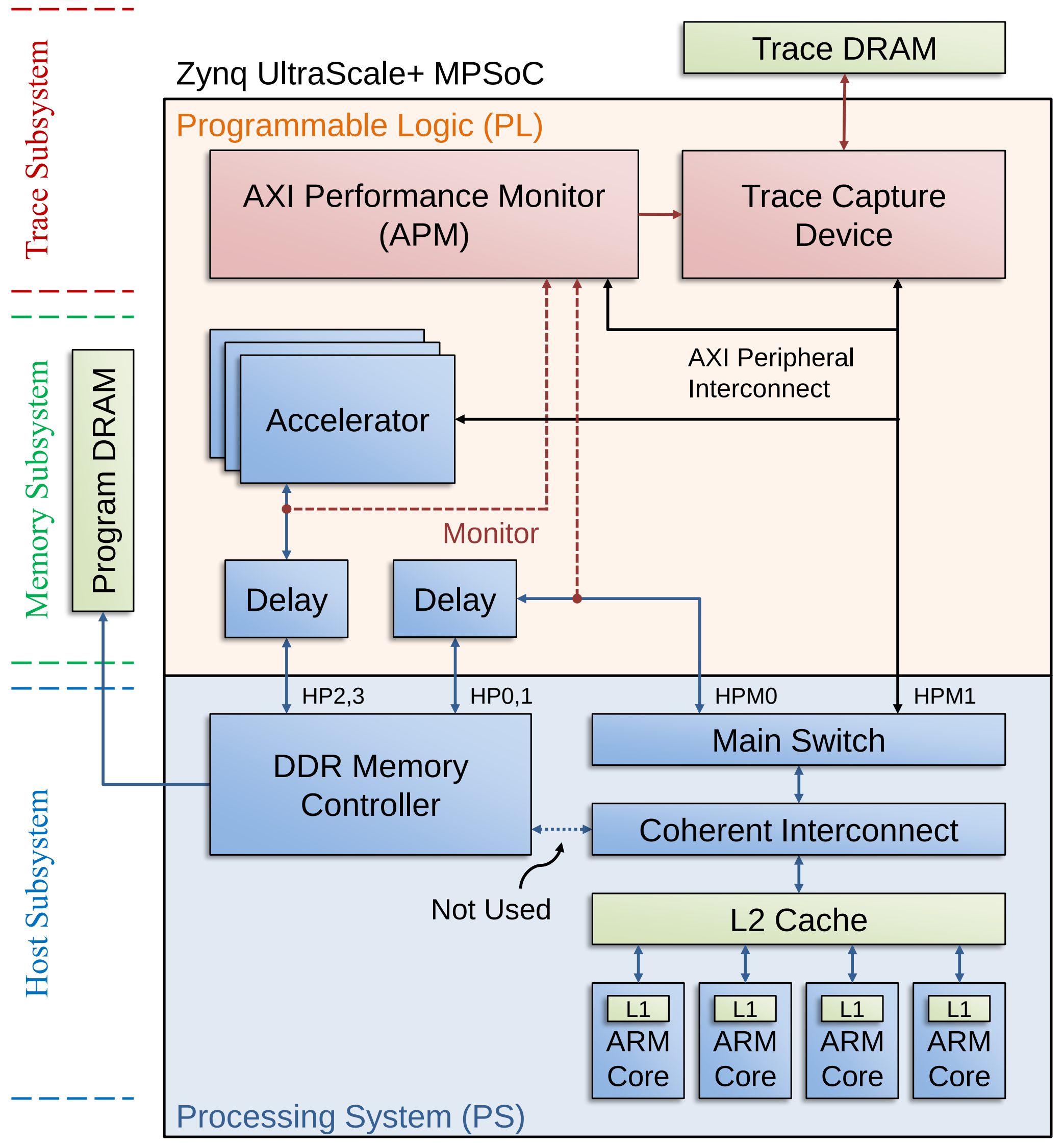


HBM



3D XPoint

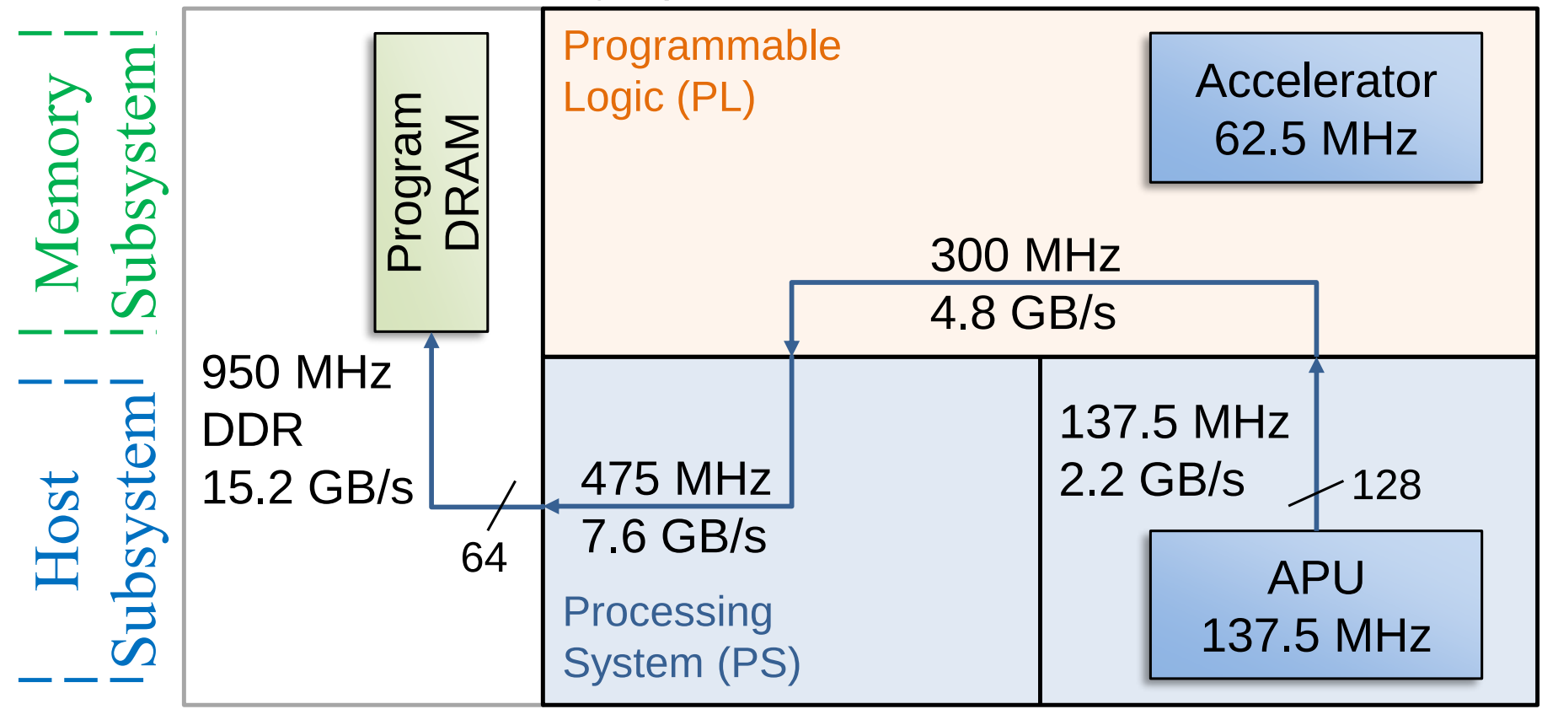




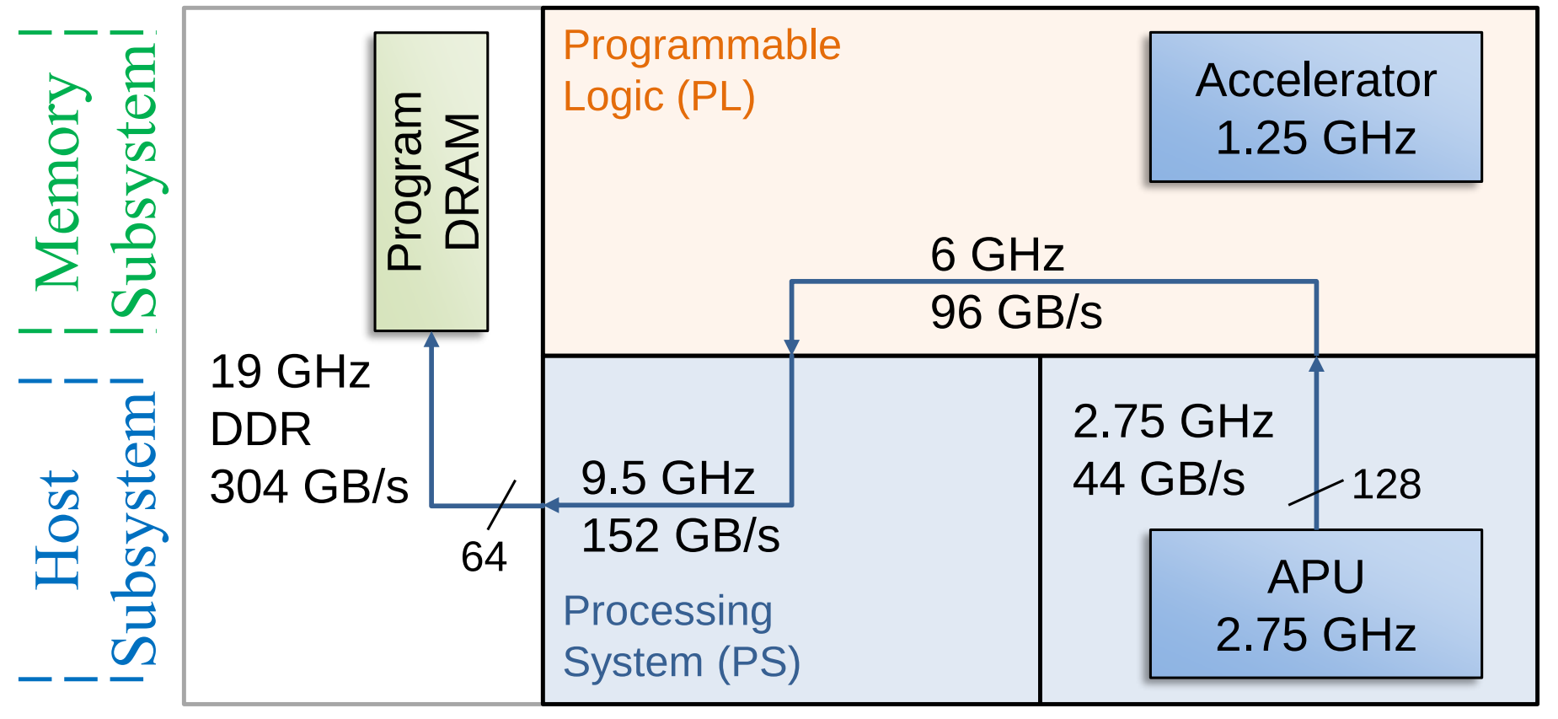
Logic in Memory Emulator (LiME)

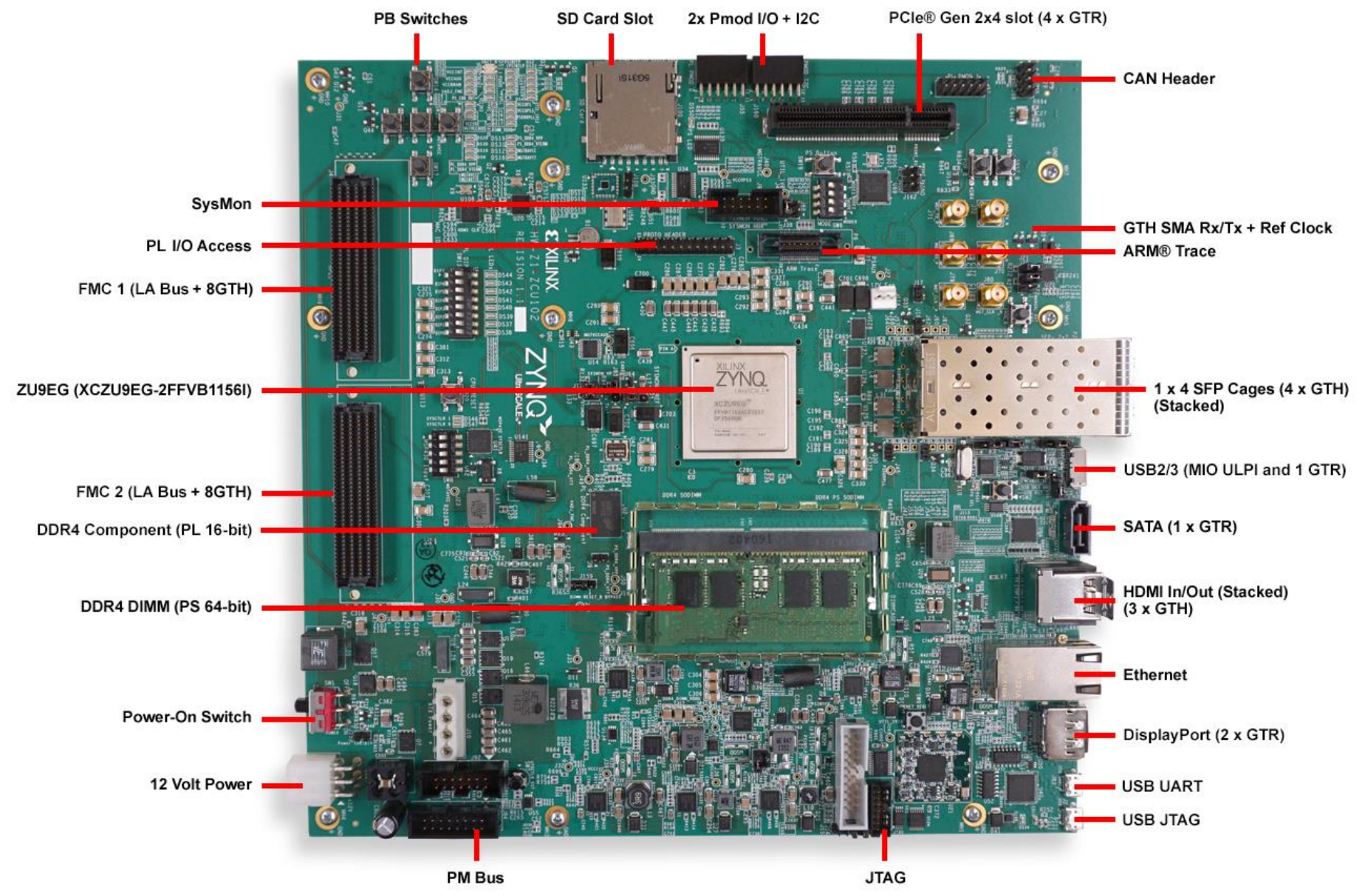
- Route memory traffic through hardware IP blocks deployed in the programmable logic
- Non-intrusively record memory transactions generated by an application
- Run applications with a slowdown of only 20x from real time
- Configurable memory subsystem latency from 10 ns to 174 us in 0.16 ns increments
- Enable tracing and statistics gathering only in regions of interest. This reduces the amount of data captured during analysis.

Zynq UltraScale+: Actual



Zynq UltraScale+: Emulated at 20x



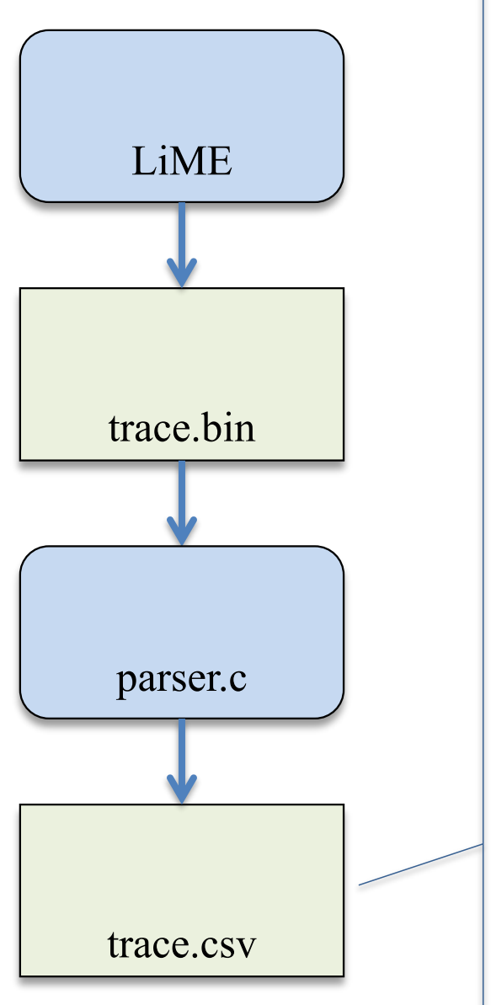


Xilinx ZCU102 Development Board

Component	Actual	Emulated
Memory Bandwidth (PL)	4.8 GB/s	96 GB/s
Memory Latency (PL)	230 ns	12 ns (too low)
Memory Latency (PL) w/delay	230 ns	12+88 = 100 ns
CPU Frequency	137.5 MHz	2.75 GHz
CPU Bandwidth	2.2 GB/s	44 GB/s
Accelerator Frequency	62.5 MHz	1.25 GHz
Accelerator Bandwidth	Up to 4.8 GB/s	Up to 96 GB/s

Use-Case: Memory trace capture and logging

- Memory traces include the address, length and timestamp for each event
- Time-stamped memory traces can be replayed on a different memory system to study bank conflict, strided access patterns, and dependency chains such as pointer chasing
- C support library provides simple macros to turn a memory trace on and off

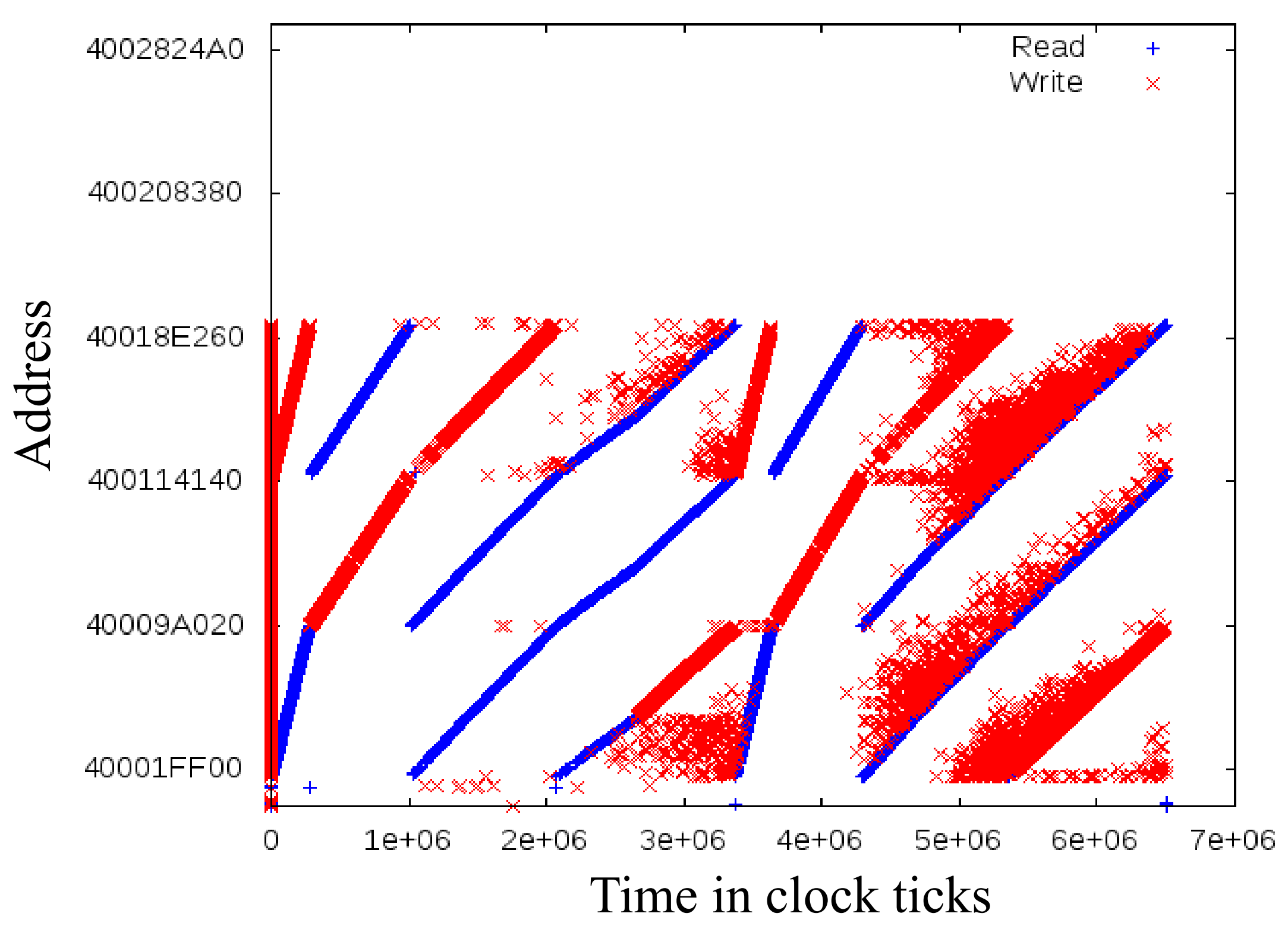


Source	Type	Address	Length	AXI ID	Time
0	W	0x400082F80	64	525	481545
0	W	0x400082FC0	64	525	482101
0	R	0x4002011C0	64	1165	482432
0	R	0x400201200	64	1293	482441
0	R	0x400201240	64	1037	487379
0	R	0x400201280	64	1037	492539
1	W	0x400080000	8	3	498523
1	R	0x400082000	16	0	493495
1	W	0x400080008	8	3	498557
1	W	0x400080010	8	3	503270
1	W	0x400080018	8	3	503304
1	W	0x400080020	8	3	503400

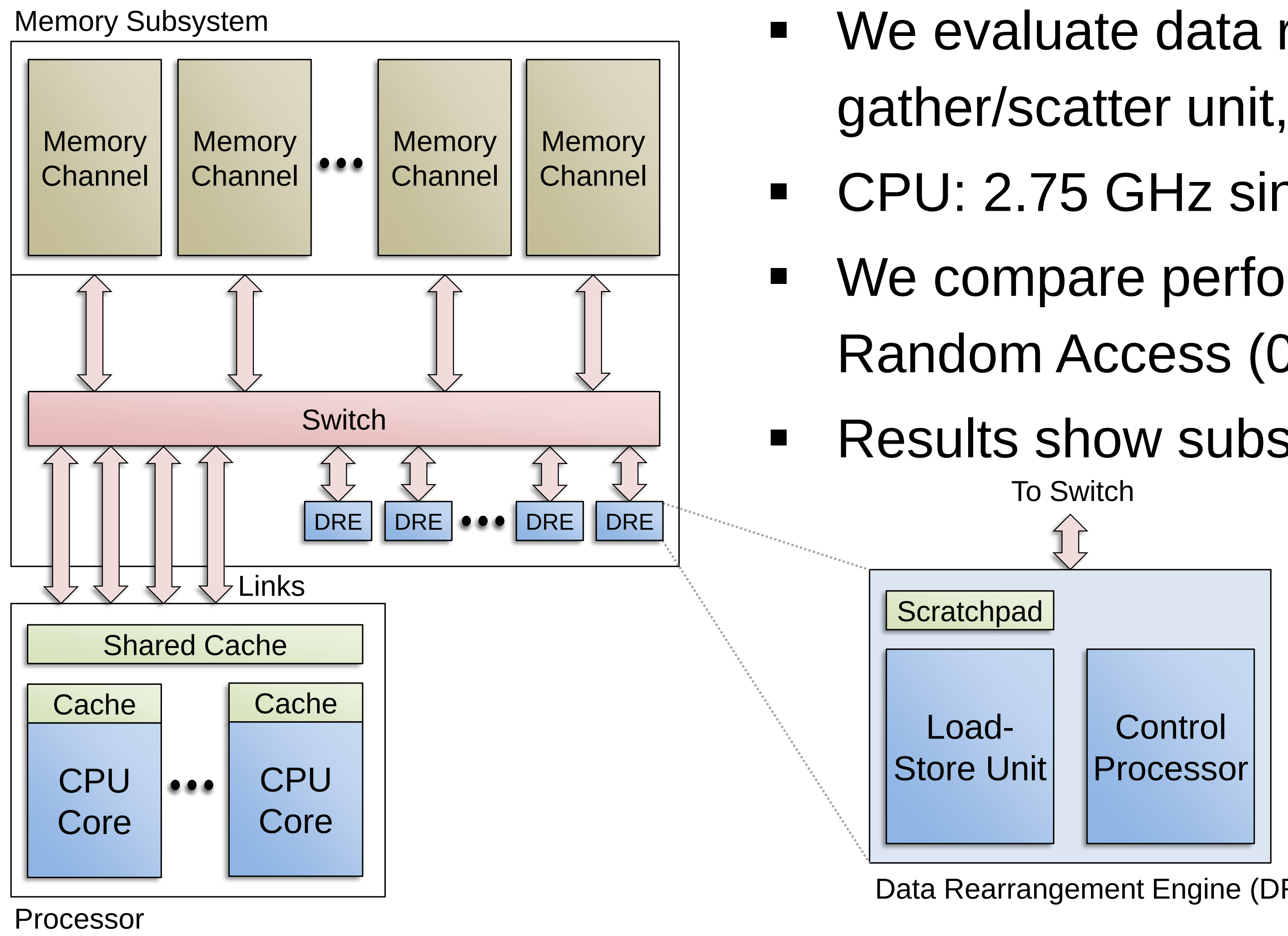
CPU = 0, Accelerator = 1

Each count represents 0.16 ns

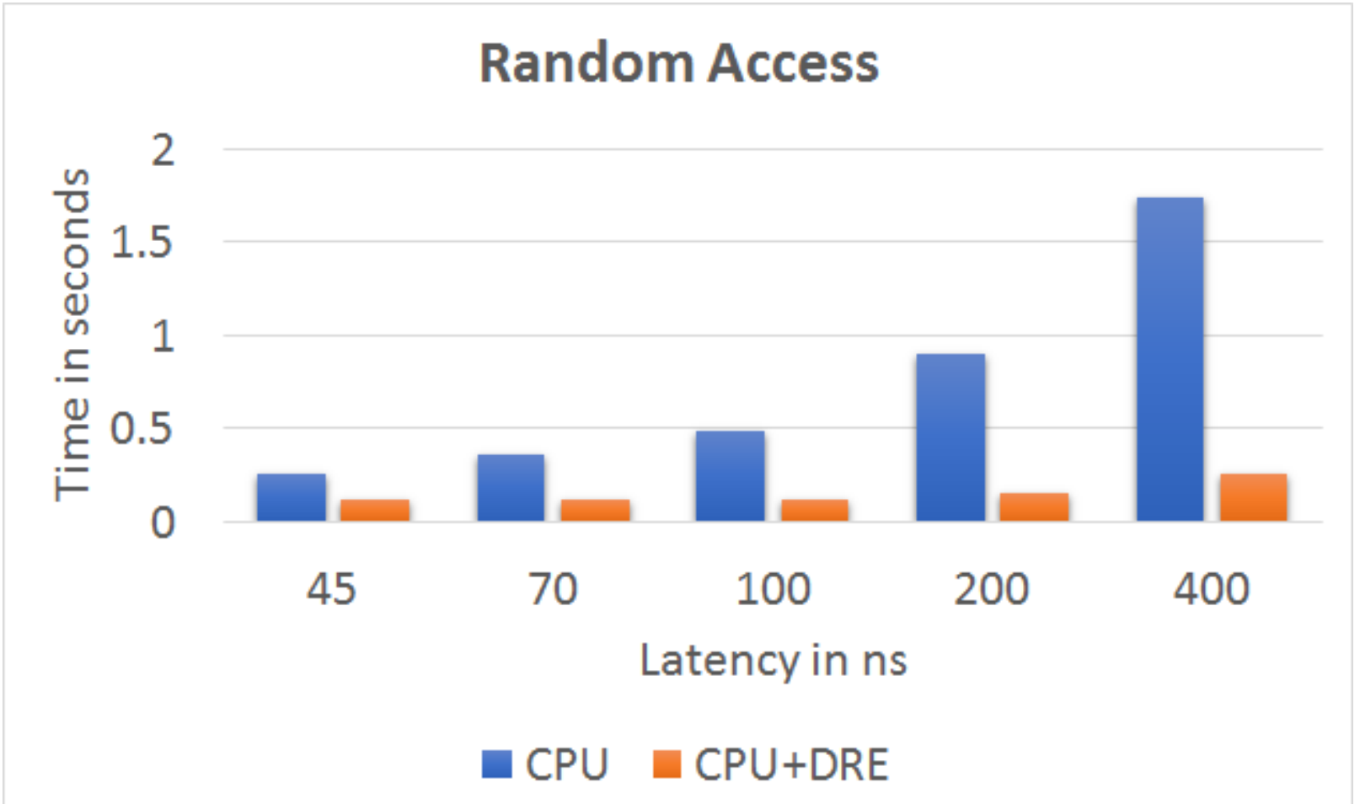
STREAM Benchmark



Use-Case: Evaluation of near-memory accelerators



- We evaluate data rearrangement engine (DRE), basically a gather/scatter unit, collocated with a memory subsystem
- CPU: 2.75 GHz single core processor, DRE runs at 1.25 GHz
- We compare performance of CPU-only with CPU+DRE for Random Access (0.5 GB size table and 4M updates)
- Results show substantial speedup using a DRE



1. A. K. Jain, G. S. Lloyd, and M. B. Gokhale. "Microscope on Memory: MPSoC-enabled Computer Memory System Assessments." FCCM 2018
2. G. S. Lloyd, and M. B. Gokhale. "In-memory data rearrangement for irregular, data-intensive computing.“ IEEE Computer 2015
3. LiME Open Source Release for ZC706 Platform: https://bitbucket.org/PerMA/emulator_st/

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-POST-752860