NEAR DATA PROCESSING: ARE WE THERE YET?

MAYA GOKHALE

LAWRENCE LIVERMORE NATIONAL LABORATORY

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. DE-AC52-07NA27344.
LLNL-PRES-665279

OUTLINE

Why we need near memory computing

Niche application

Data reorganization engines

Computing near storage

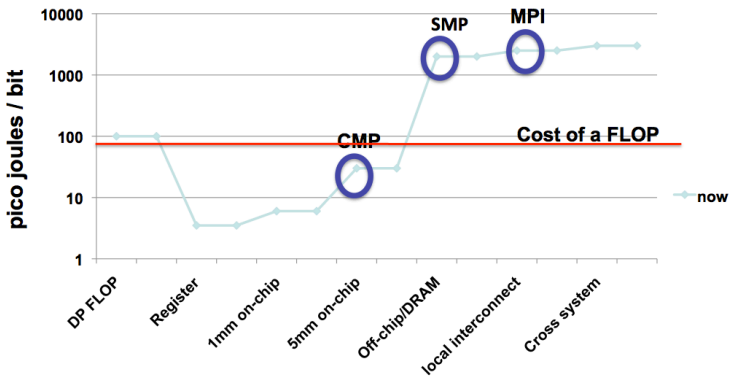
FPGAs for computing near memory

WHY AREN'T WE THERE YET?

- **Processing near memory is attractive for high bandwidth, low latency, massive parallelism**
- **90's era designs closely integrated logic and DRAM transistor**
  - Expensive
  - Slow
  - Niche
- **Is 3D packaging the answer?**
  - Expensive
  - Slow
  - Niche
- **What are the technology and commercial incentives?**

EXASCALE POWER PROBLEM: DATA MOVEMENT IS A PRIME CULPRIT

- Cost of a double precision flop is negligible compared to the cost of reading and writing memory
- Cache-unfriendly applications waste memory bandwidth, energy
- Number of nodes needed to solve problem is inflated due to poor memory bw utilization
- Similar phenomenon in disk



Operation	Energy (pJ)
64-bit integer operation	1
64-bit floating-point operation	20
256 bit on-die SRAM access	50
256 bit bus transfer (short)	26
256 bit bus transfer (1/2 die)	256
Off-die link (efficient)	500
256 bit bus transfer(across die)	1,000
DRAM read/write	16,000
HDD read/write	$O(10^6)$

Sources: Horst Simon, LBNL
Greg Astfalk, HP

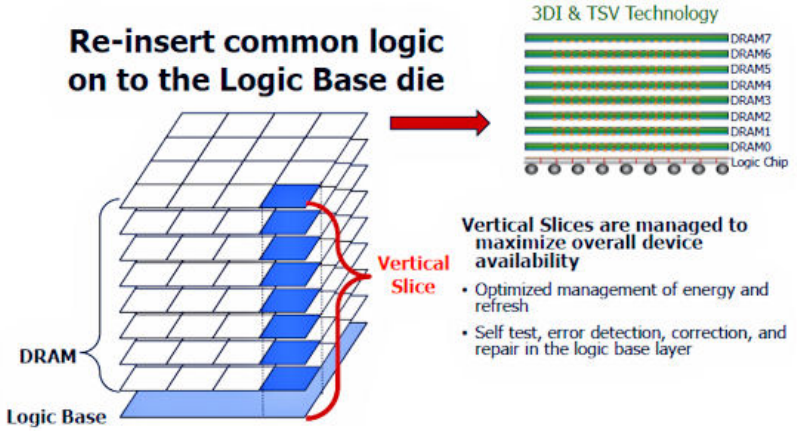
MICRON HYBRID MEMORY CUBE OFFERS OPPORTUNITY FOR PROCESSING NEAR MEMORY

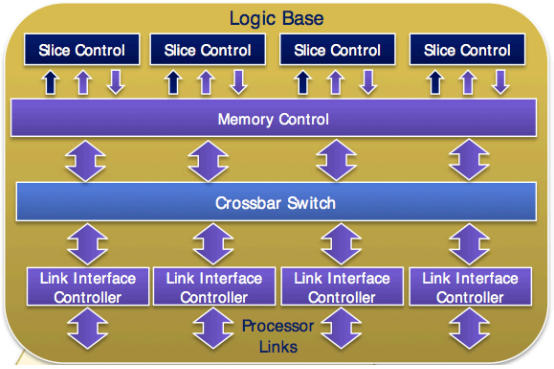
Fast logic layer

- Customized processors
- Through silicon vias (TSV) for high bandwidth access to memory

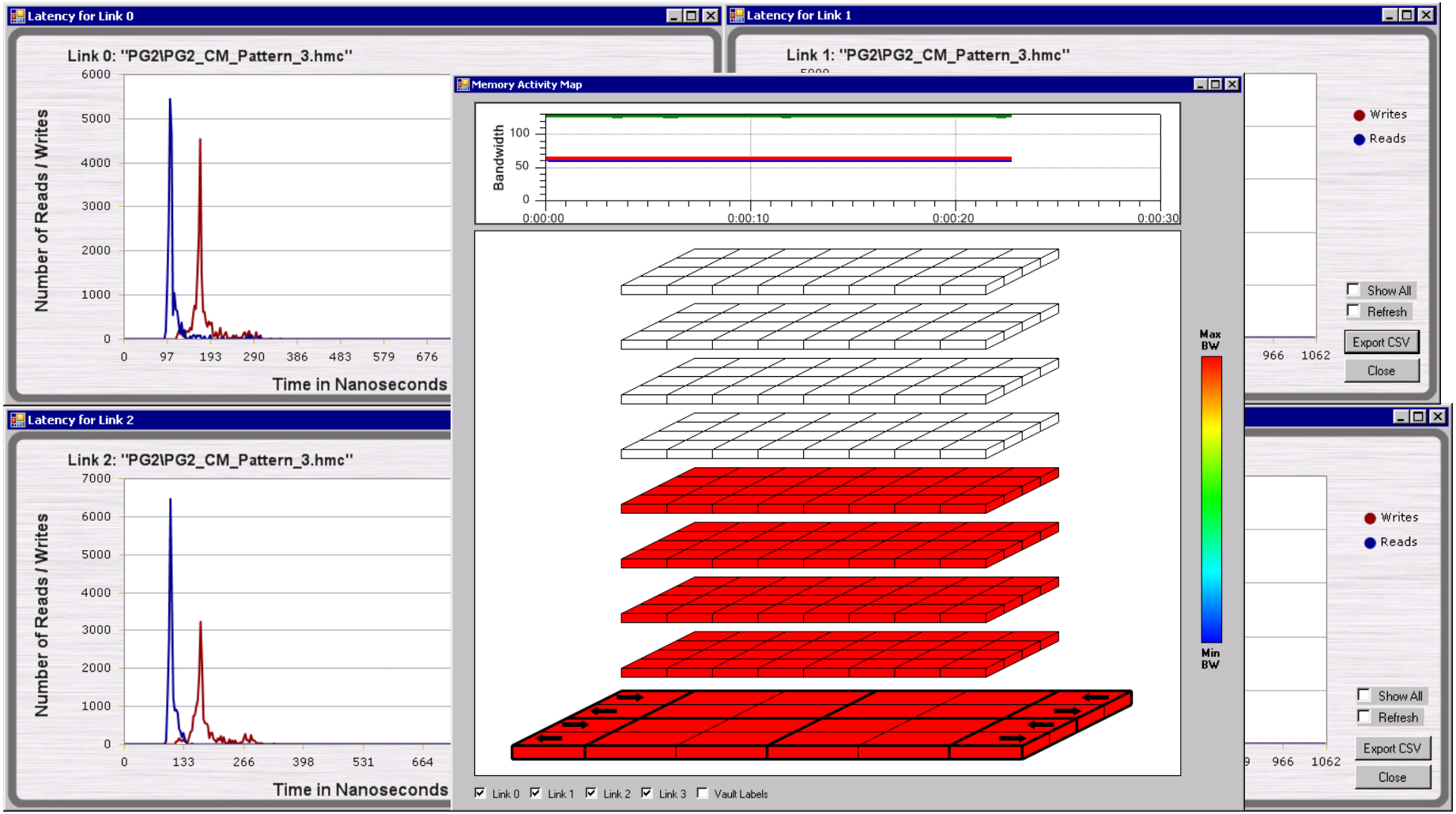
Vault organization yields enormous bandwidth in the package

Best case latency is higher than traditional DRAM due to packetized abstract memory interface

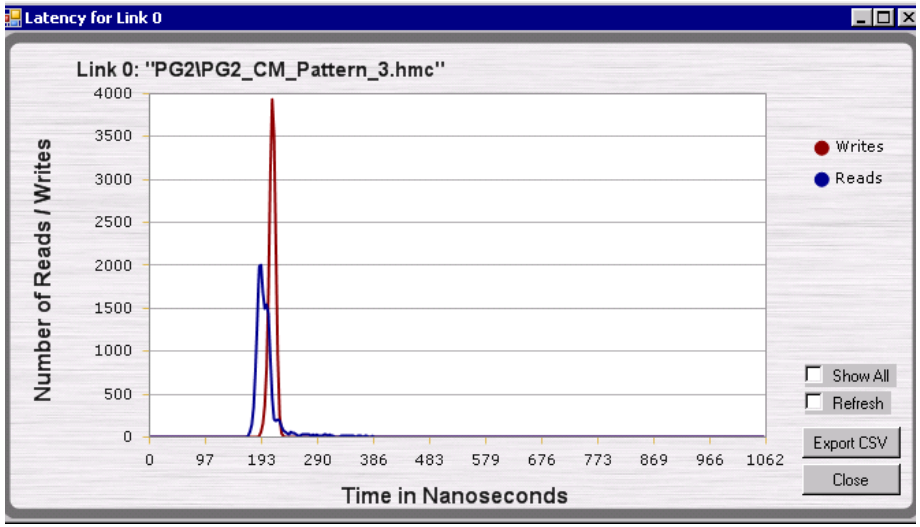


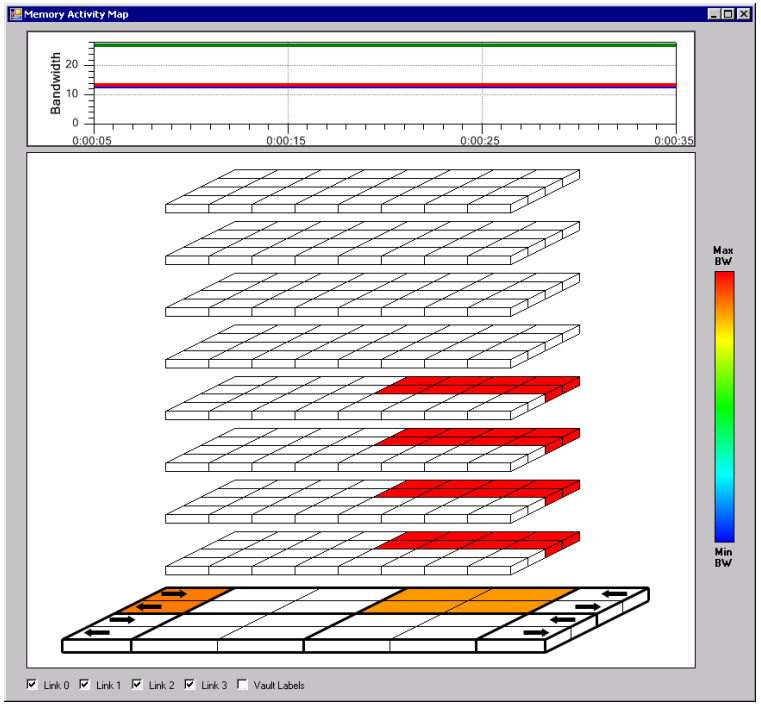


HMC LATENCY: LINK I TO VAULT I, 128B, 50/50



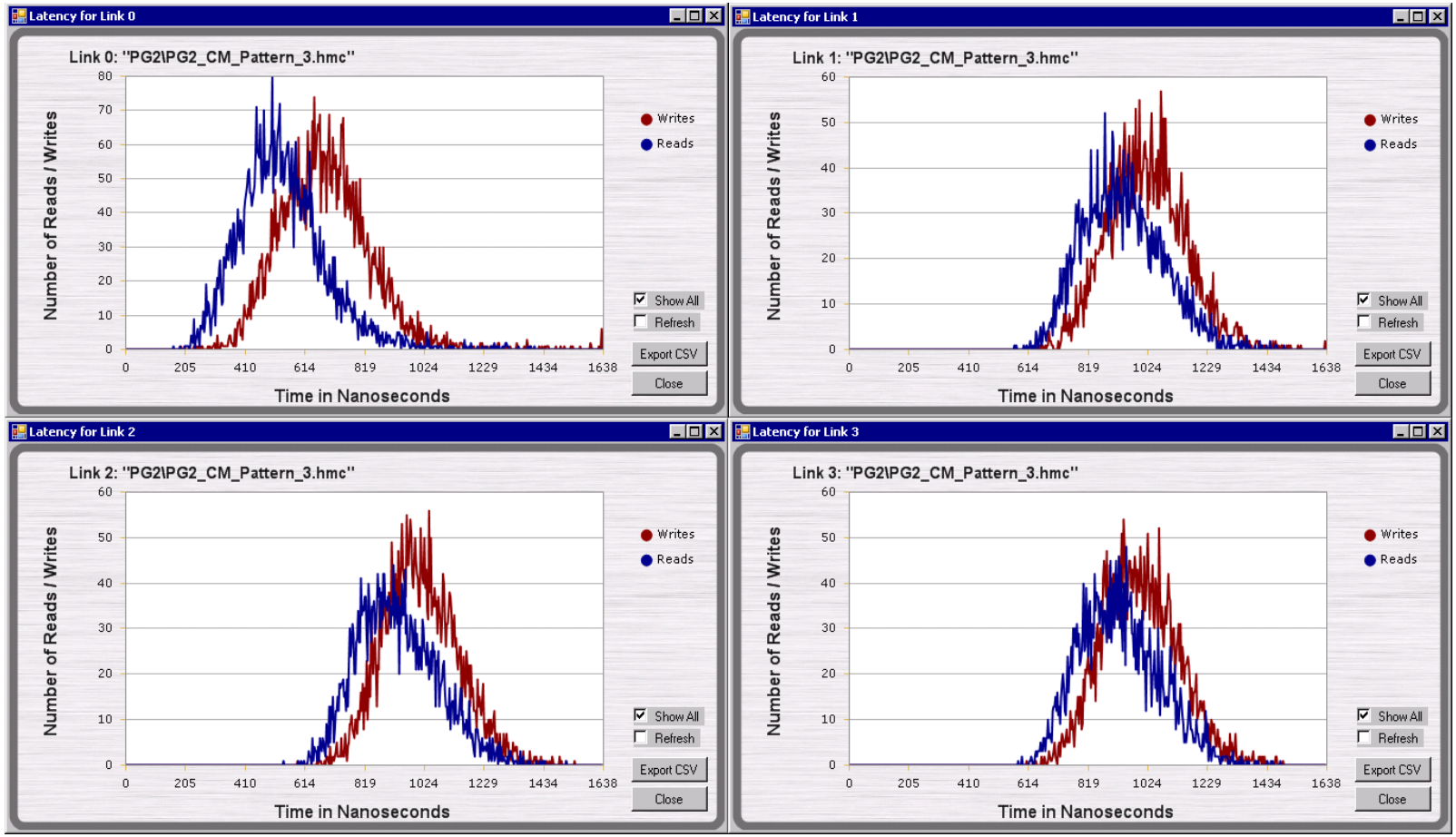
HMC LATENCY: LINK I TO VAULT J, 128B, 50/50





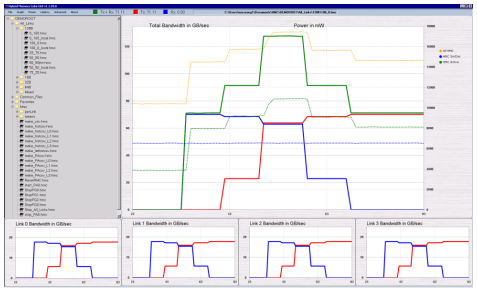
One link active, showing effect of request through communication network.

HMC: WIDE VARIABILITY IN LATENCY WITH CONTENTION

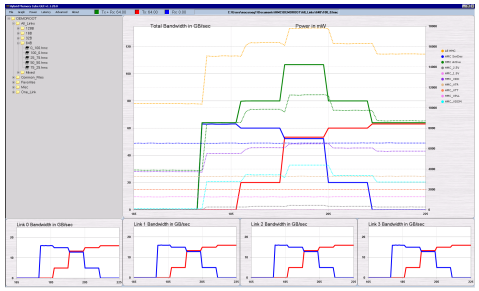


Requests on links 1-4 for data on vaults 0-4 (link 0 local)

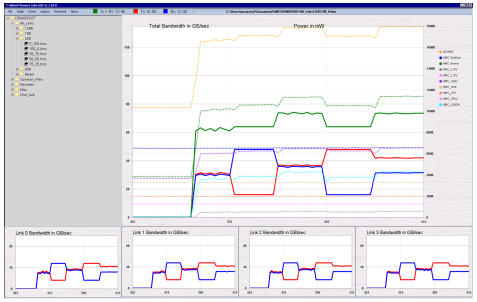
HMC SHOWS HIGH BANDWIDTH FOR REGULAR, STREAMING ACCESS PATTERNS, BUT …



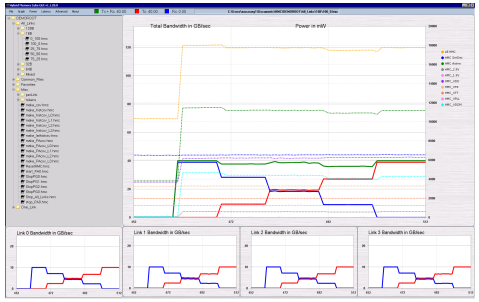
128B payload



64B payload



32B payload



16B payload

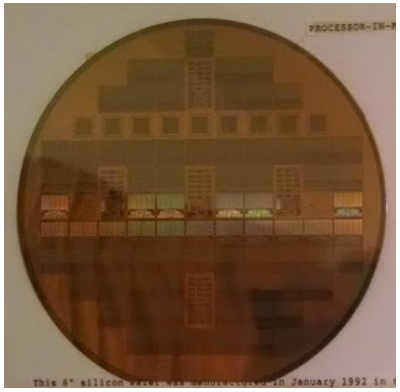
Small payload, irregular accesses cut bandwidth by factor of 3

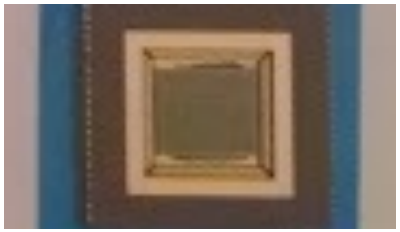
SPECIALIZED PROCESSING FOR HIGH VALUE APPLICATIONS

Processing in memory/storage might make sense for niche, high value applications

- **Image pixel processing, character processing (bioinformatics applications)**
- **Massively parallel regular expression applications (cybersecurity, search problems cast as RE)**
- **Content addressable memories**

PIXEL/CHARACTER PROCESSING IN MEMORY ARCHITECTURE





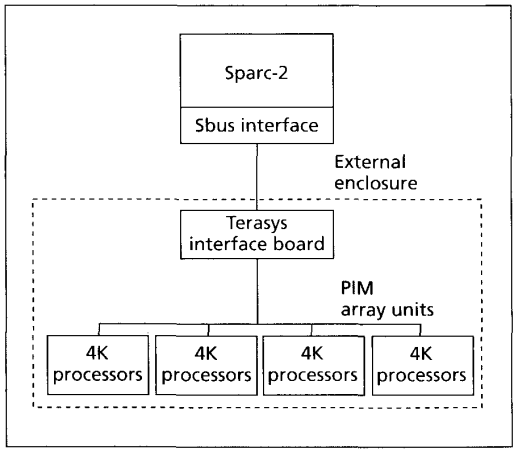


Figure 1. A 16K processor Terasys workstation.

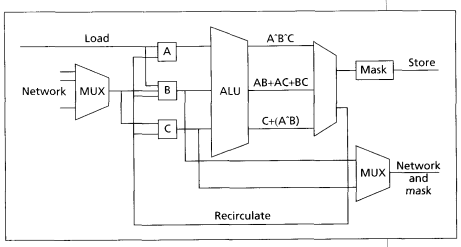


Figure 3. A processor-in-memory processor.

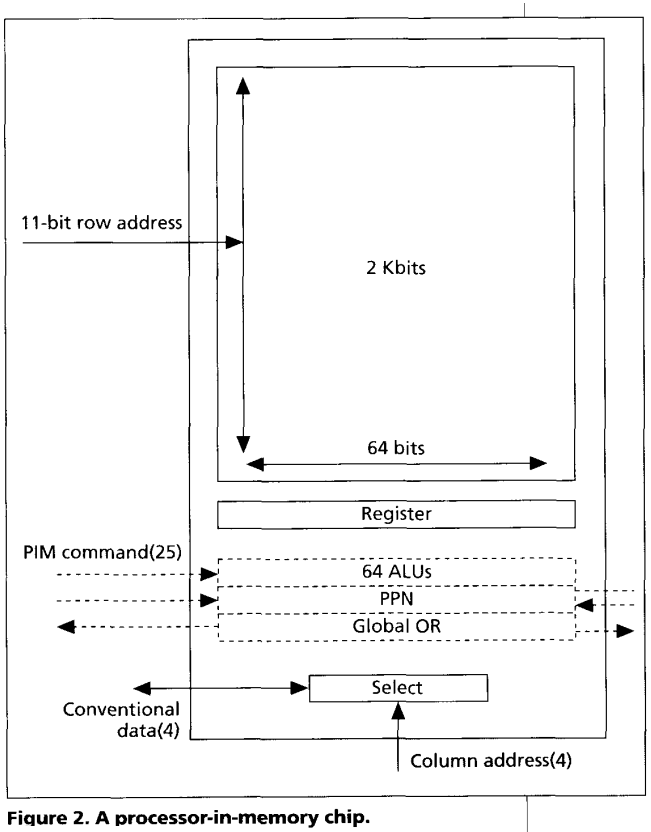


Figure 2. A processor-in-memory chip.

```
poly unsigned x:4;   /* 4-bit logical variable x */
typedef unsigned poly int33:33;   /* 33-bit logical, user-defined type */
poly int y: 1000;   /* arbitrarily large variables are supported */
typedef poly struct
  {  int33 A[50];
     int33 B;
     char c;
  }  S;   /* poly structure */
```

Figure 4. Parallel variables in dbC (data-parallel bit

```
poly x:10, y:11
x[4:8] = y[0:4];        /* start/end index */
x[4+:5] = y[0+:5];      /* bit length notation */
```

Figure 5. Bit insertion and extraction in dbC.

NFA COMPUTING IN DRAM: MASSIVE PARALLELISM

Micron's automata processor puts processing in the DRAM

- Massively parallel 2D fabric
- Thousands – millions of automaton processing elements "state transition elements"
  - Implement Non-deterministic finite automata
  - Augmented with counters
  - Functions can be reprogrammed at runtime
- Routing matrix consists of programmable switches, buffers, routing wires, cross-point connections
  - Route outputs of one automaton to inputs of others
  - Possible multi-hop
- DDR-3 interface
- Estimate 4W/chip; snort rules in 16W vs 35W for FPGA

AUTOMATA PROCESSOR APPLICATIONS RELATE TO REGULAR EXPRESSIONS

Each chip takes an 8-bit symbol as input

8➔256 decode for 256-bit "row buffer" and each column is 48K

Can match regular expression, with additional counting

- $a^+b^+c^+$, total # symbols = 17
- Cybersecurity, bioinformatics applications

Challenge is to expression other problems as NFAs

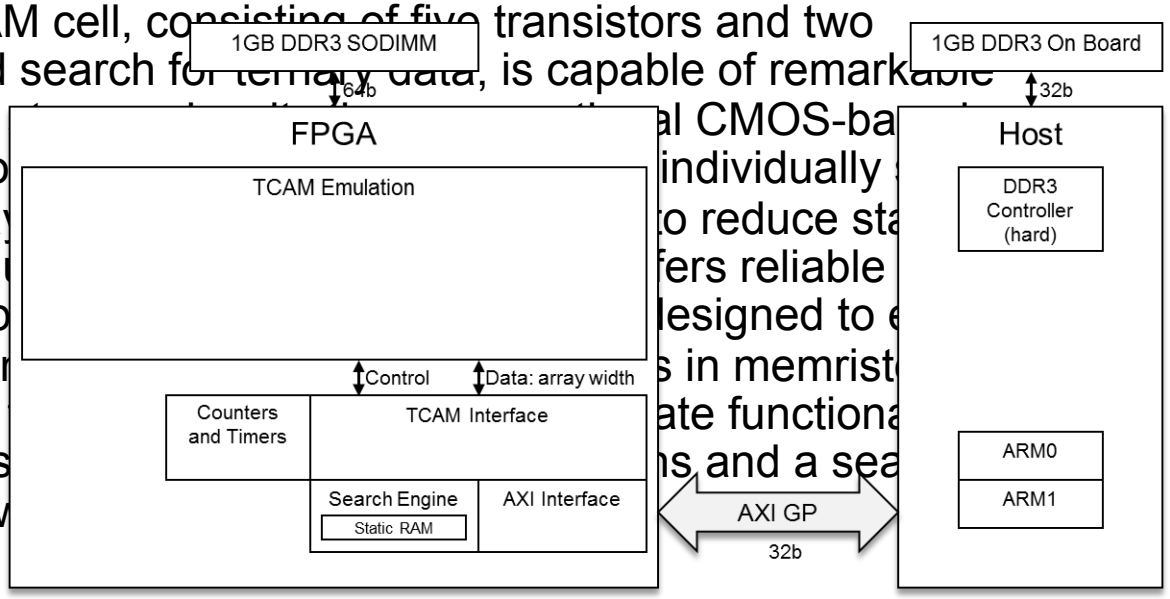
- Graph search

TCAM IN MEMRISTOR FOR CONTENT BASED SEARCH

Memristor-based ternary content addressable memory (mTCAM) for data-intensive computing (Semiconductor Science and Technology. V. 29, no. 10, 2014,

http://stacks.iop.org/0268-1242/29/i=10/a=104010

A memristor-based ternary content addressable memory (mTCAM) is presented. Each mTCAM cell, consisting of five transistors and two memristors to store and search for ternary data, is capable of remarkable nonvolatility and higher al CMOS-ba TCAMs. Each memristo individually high impedance is alwa o reduce sta energy consumption. A fers reliable energy-efficient write op esigned to optimum sensing margi s in memrist devices. Simulations of ate functiona write and search modes ns and a sea 0.99 fJ/bit/search for a w



DATA ANALYTICS: HIGH VALUE COMMERCIAL APPLICATIONS

Application space includes …

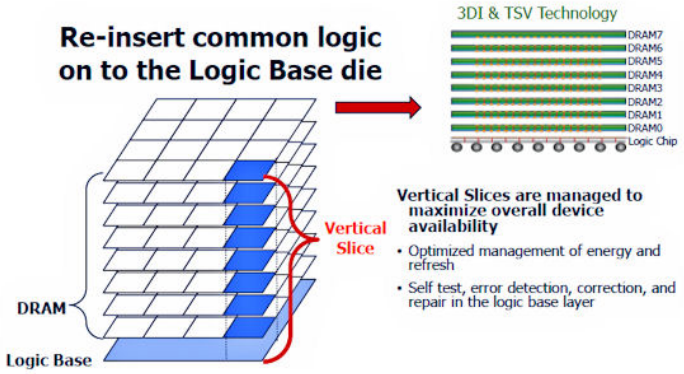
- **Image/video processing**
- **Graph analytics**
- **Specialized in-memory data bases**
- **Tree, list-based data structure manipulation**

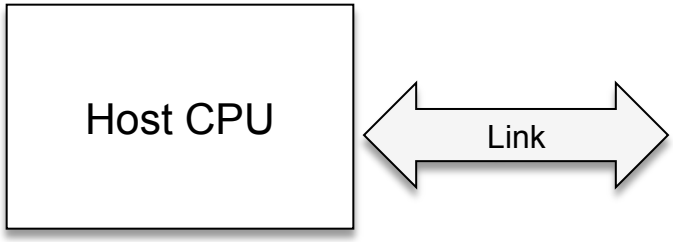
Minimal benefit from cache

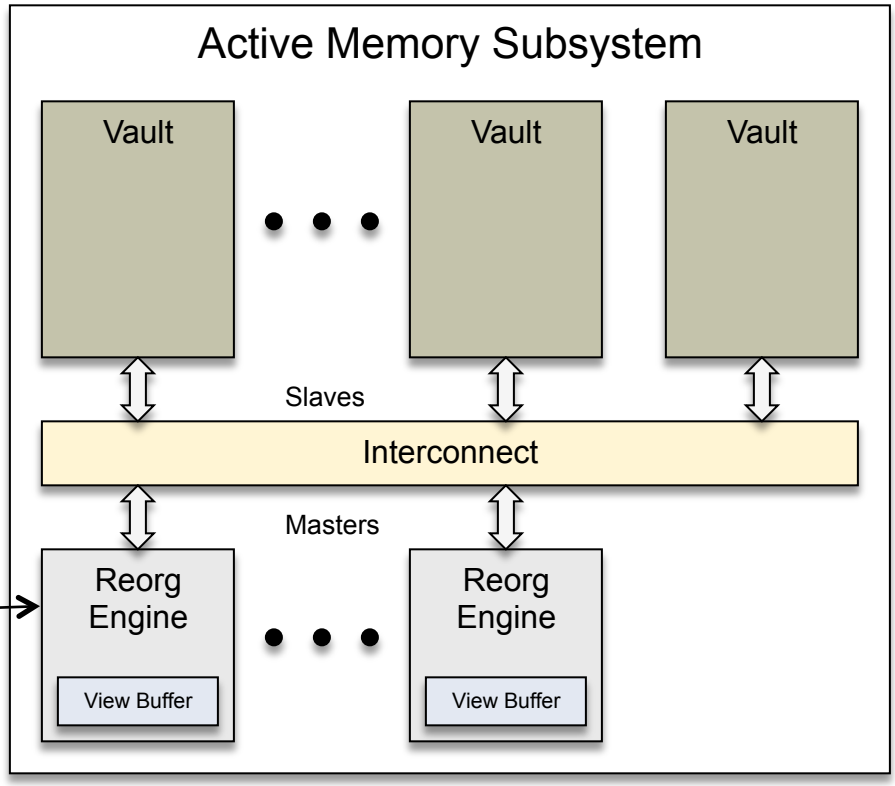
- **Streaming, strided access patterns**
- **Irregular access patterns**

Processing in (near) memory opportunity to "reorganize" data structures from native format to cache-friendly

PROPOSED DATA REORGANIZATION ENGINE ARCHITECTURE







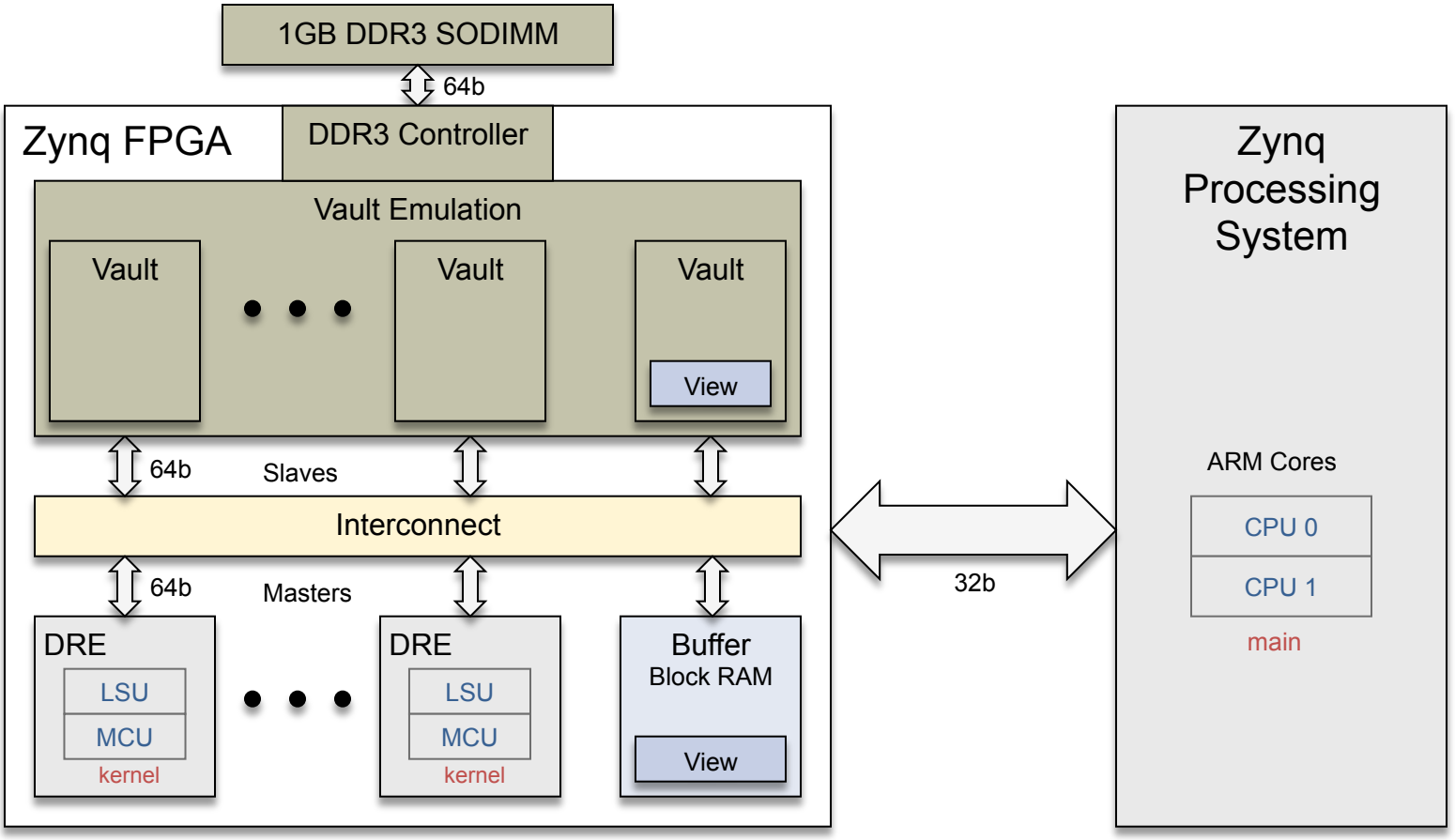
Data Reorganization Engines

- Multiple block requests given at start
- Incoming packets for Data Reorg Engine processed one at a time
- All reorganization done within Engine
- Views are created in buffers
- View buffers are memory mapped from host CPU perspective

FPGA emulator for Data Reorg Engine in active memory

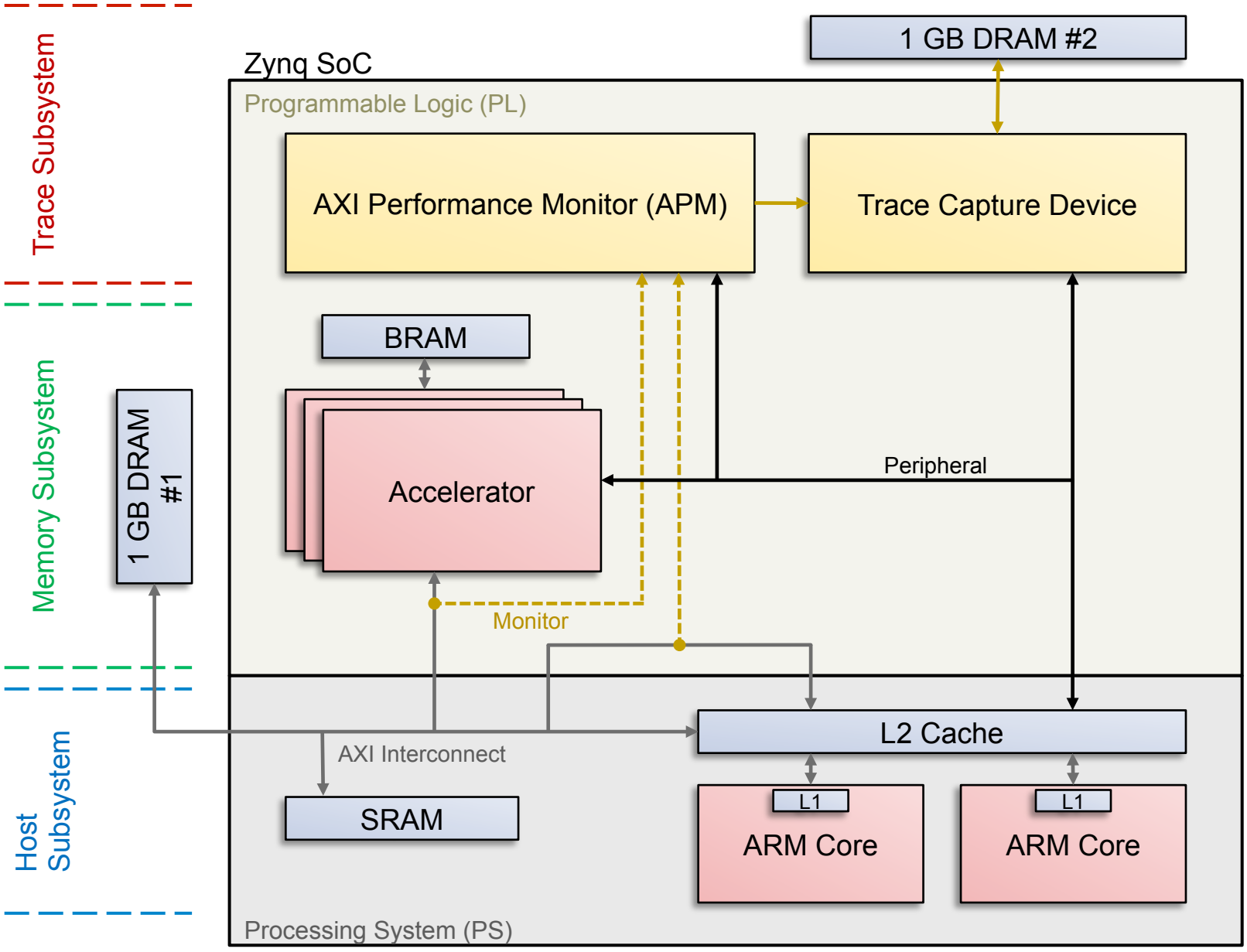
SCOTT LLOYD, LLNL

EMULATION ARCHITECTURE



The Zynq system on a chip with both hard processors and FPGA logic is used for active memory emulation. The ARM cores (CPU) are used for the main application and the kernel executes on a data reorganization engine (DRE) which consists of a micro-control unit (MCU) and a load-store unit (LSU). The vaults are emulated with DDR3 memory.

EMULATOR STRUCTURE ON ZYNQ



DATA CENTRIC BENCHMARKS DEMONSTRATE PROGRAMMABLE DRES

Image difference: compute pixel-wise difference of two reduced resolution images

- DRE in memory subsystem assembles view buffer of each reduced resolution image
- Host CPU computes image difference

Pagerank: traverse web graph to find highly referenced pages

- DRE in memory subsystem assembles view buffer of node neighbors (page rank vector)
- Host CPU updates reference count of each neighbor, creates next page rank vector

RandomAccess: use memory location as index to next location

- DRE in memory subsystem assembles view buffer of memory words accessed through gather/scatter hardware

EMULATOR CONFIGURATION

Emulated System Configuration

CPU: 32-bit ARM A9 core, 2.57 GHz, 5.14 GB/s bandwidth

DRAM: 1 GB, 122 ns read and write access

SRAM: 256 KB, 42 ns read and write access

----- Data Reorganization Engine (DRE) in memory subsystem -----

LSU: (Load-Store Unit) 64-bit data path, 1.25 GHz, 10.00 GB/s bandwidth

DRAM: 1 GB, 90 ns read and write access

SRAM: 256 KB, 10 ns read and write access

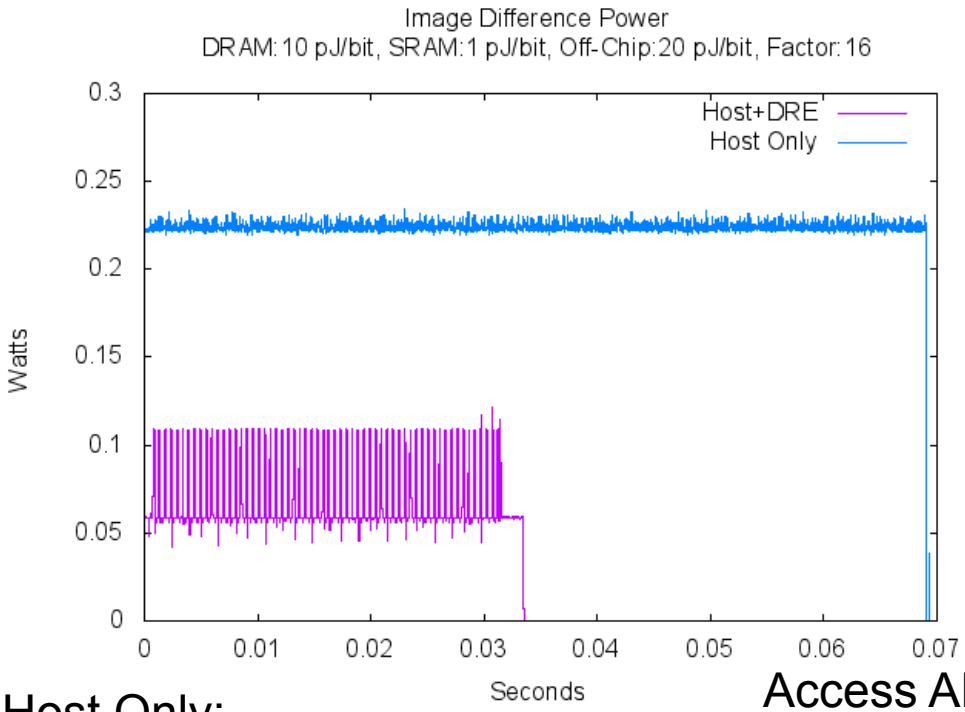
MCU: (MicroBlaze) 32-bit data path, 1.25 GHz, 5.00 GB/s bandwidth

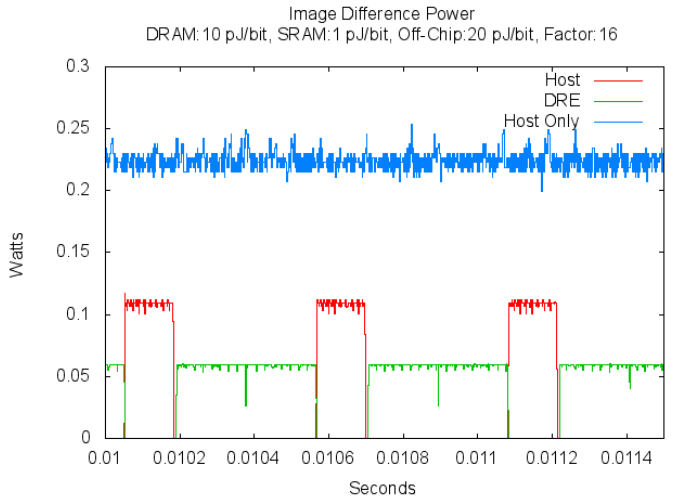
DRAM: 1 GB, 90 ns read and write access

SRAM: 256 KB, 10 ns read and write access

BRAM: local 16 KB for instructions and stack

DRE USE FASTER, LESS ENERGY: IMAGE DECIMATION BY 16

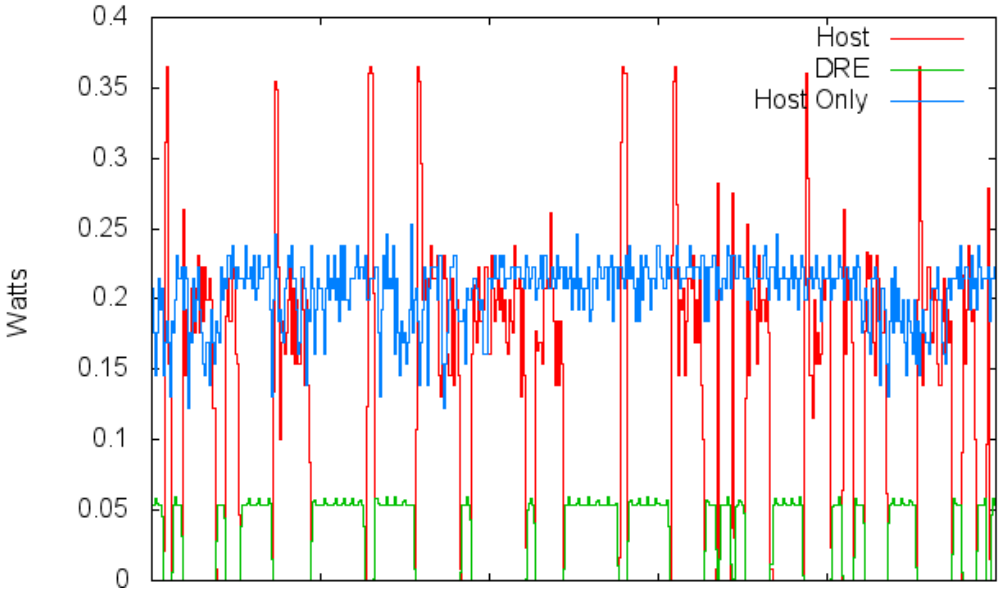




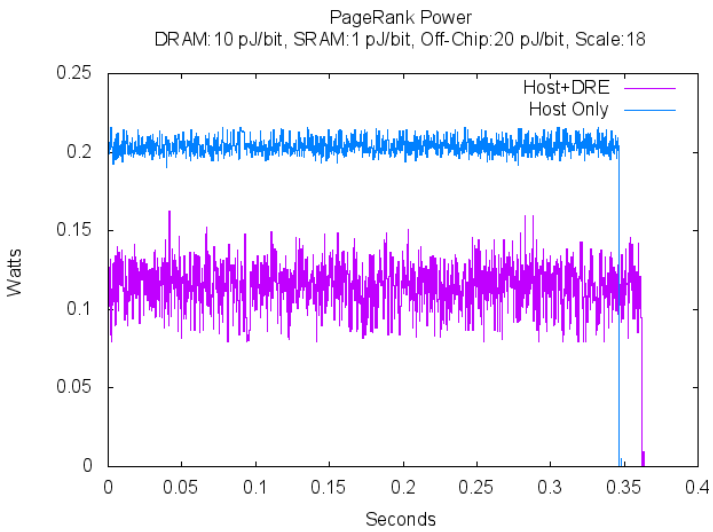
Host Only:
Access ARM read: 1997558
ARM write: 15764
Bytes ARM read: 63921856
ARM write: 504448
Joules ARM: 0.0154623

Access ARM read: 156504 ARM write: 15666
Access DRE read: 1000000 DRE write: 1000000
Bytes ARM read: 5008128 ARM write: 501312
Bytes DRE read: 16000000 DRE write: 8000000
Joules ARM: 0.000998247
Joules DRE: 0.001344
Joules A+D: 0.00234225

PAGERANK SCALE 18,19



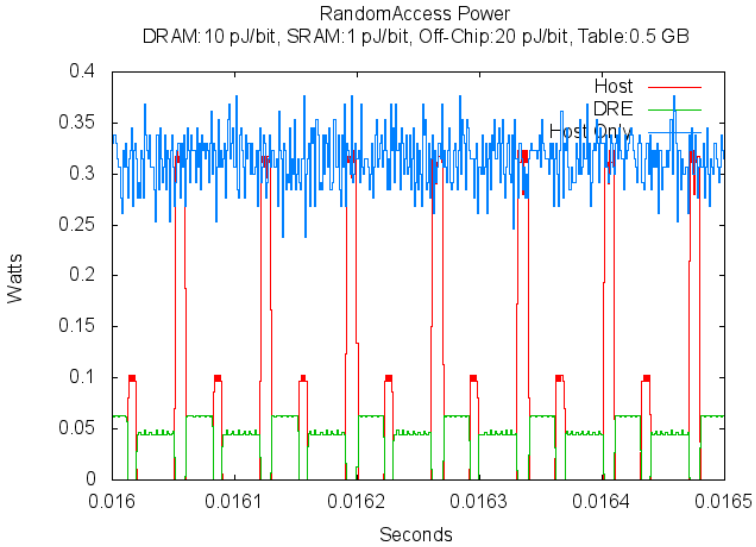
Scale 18 profile

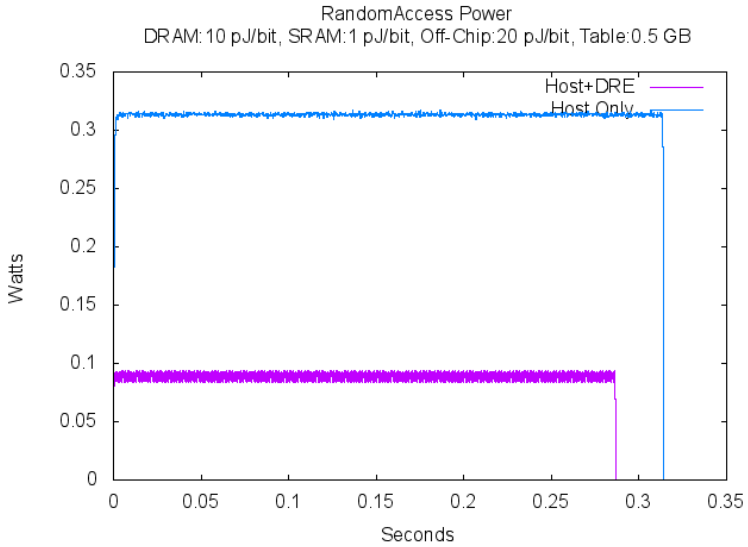


Scale 19 Host+DRE:
page rank time:0.753485 sec
Host Only:
page rank time:0.828449 sec

Profile of one iteration of the Page Rank algorithm. The DRE assembles a compact page rank view based on the adjacency list for vertex *i* which is an index array into the original page rank vector. The host accesses the compact view created by the DRE.

RANDOMACCESS





Host+DRE:
Real time used = 0.286718 seconds
0.014628678 Billion(10^9) Updates per second [GUP/s]

Host Only:
Real time used = 0.313953 seconds
0.013359660 Billion(10^9) Updates per second [GUP/s]

RESEARCH QUESTIONS REMAIN

Enough benefit to justify cost of fabrication?

How to interact with virtual memory?

How many DREs on chip?

Programming models

ACTIVE STORAGE WITH MINERVA (UCSD AND LLNL)

Designed for future generations of NVRAM that are byte addressable, very low latency on read

Based on computation close to storage

Moved data or I/O intensive computations to the storage to minimize data transfer between the host and the storage

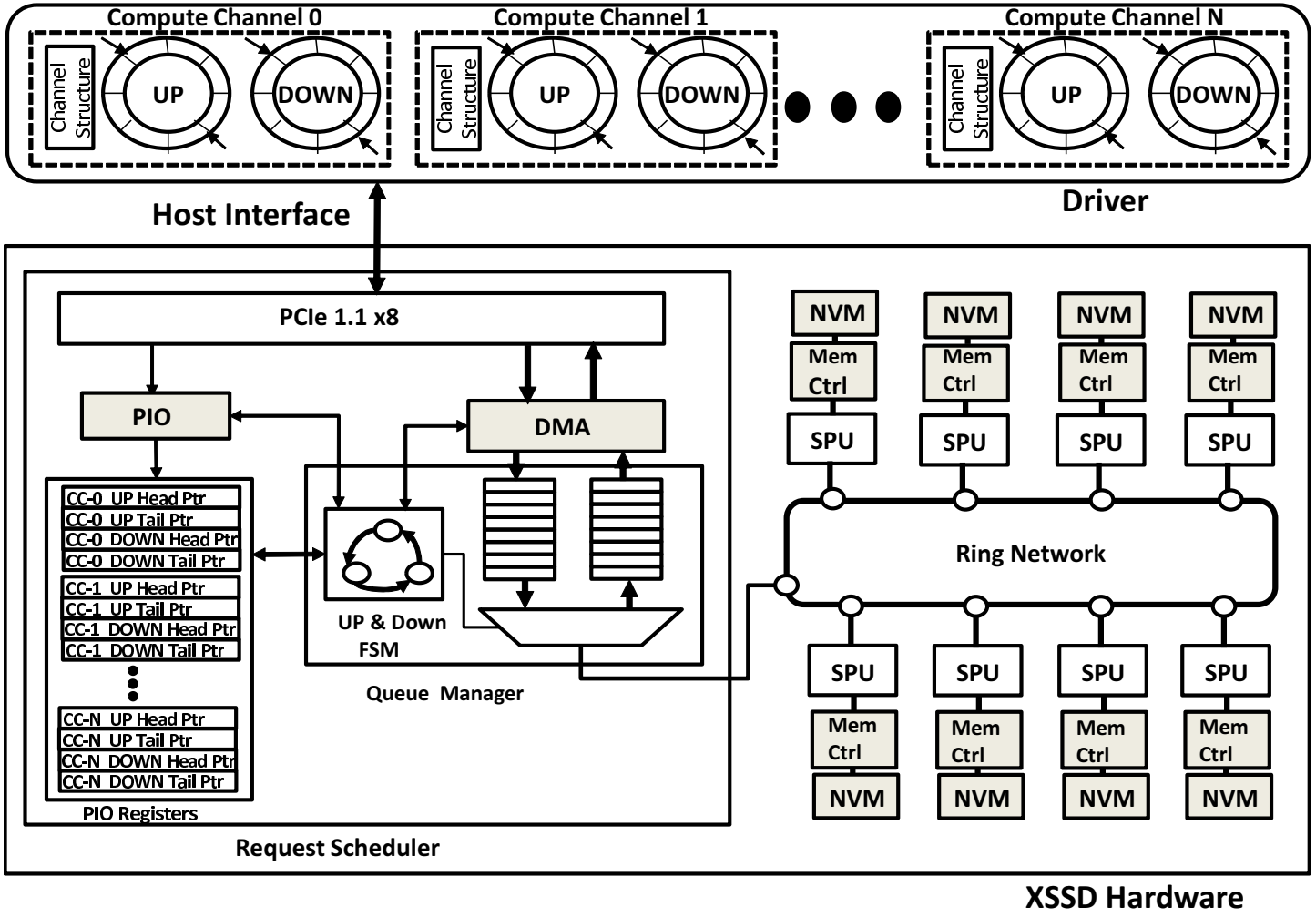
Processing scales as storage expands

Power efficiency and performance gain

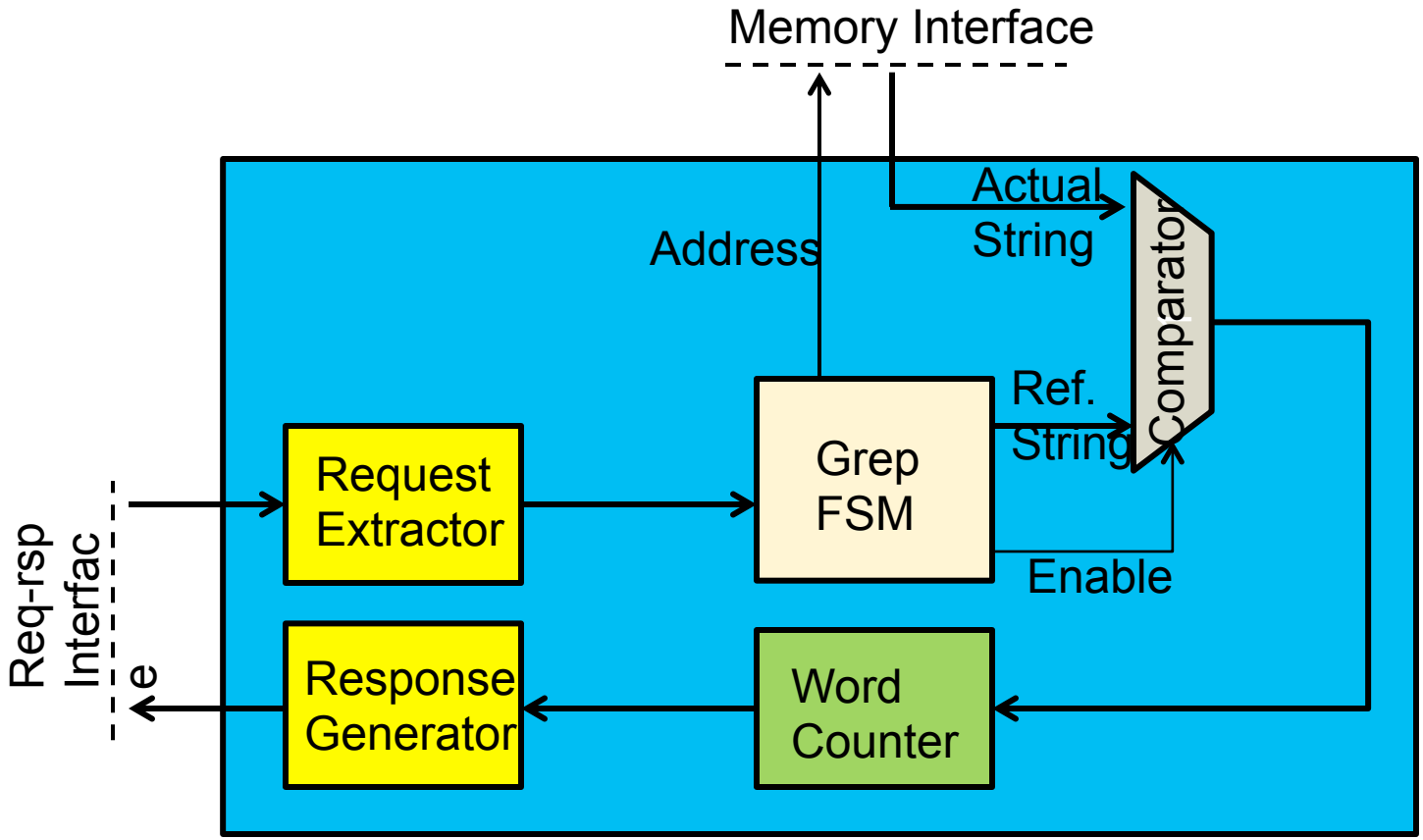
- Prototyped on BEE-3 system
  - 4 Xilinx Virtex 5, each with 16GB memory, ring interconnect
- Emulate PCM controller by inserting appropriate latency on reads and writes, emulate wear leveling to store blocks
- Get advantage of fast PCM and FPGA-based hardware acceleration

Arup De, UCSD/LLNL

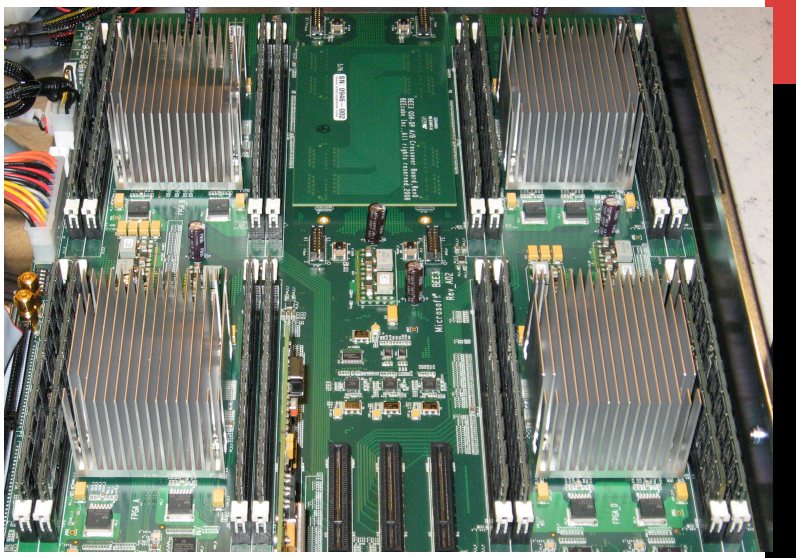
MINERVA ARCHITECTURE PROVIDES CONCURRENT ACCESS, IN-STORAGE PROCESSING UNITS



INTERACTION WITH STORAGE PROCESSOR



MINERVA PROTOTYPE



Built on BEE3 board

PCIe 1.1 x8 host connection

Clock frequency 250MHz

Virtex 5 FPGA implements

- Request scheduler
- Network
- Storage processor
- Memory controller

DDR2 DRAM emulates PCM

RESOURCE USAGE ON VIRTEX-5 LX155T

Component	Slice Regs		LUTs		BRAMs	
		%		%		%
Request Scheduler	10536	10.8	11509	11.8	60	28.3
Storage Processor	18359	18.8	14408	14.8	24	11.3
Memory Controller	6779	6.9	5405	5.5	18	8.4
PCIe-1.1 x8	3882	3.9	3008	3.0	11	5.1
Ring Network	984	1.0	856	0.8		

MINERVA FPGA EMULATOR ALLOWS US TO ESTIMATE PERFORMANCE, POWER

Compare standard I/O with computational offload approach

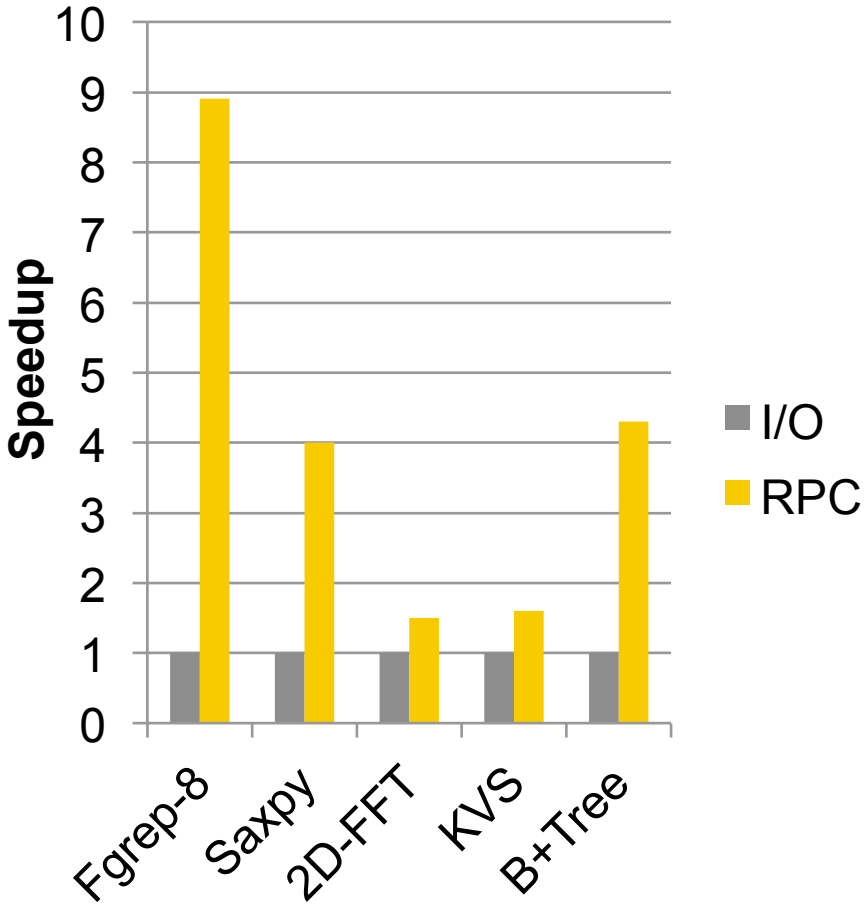
Design alternative storage processor architectures and evaluate trade-offs

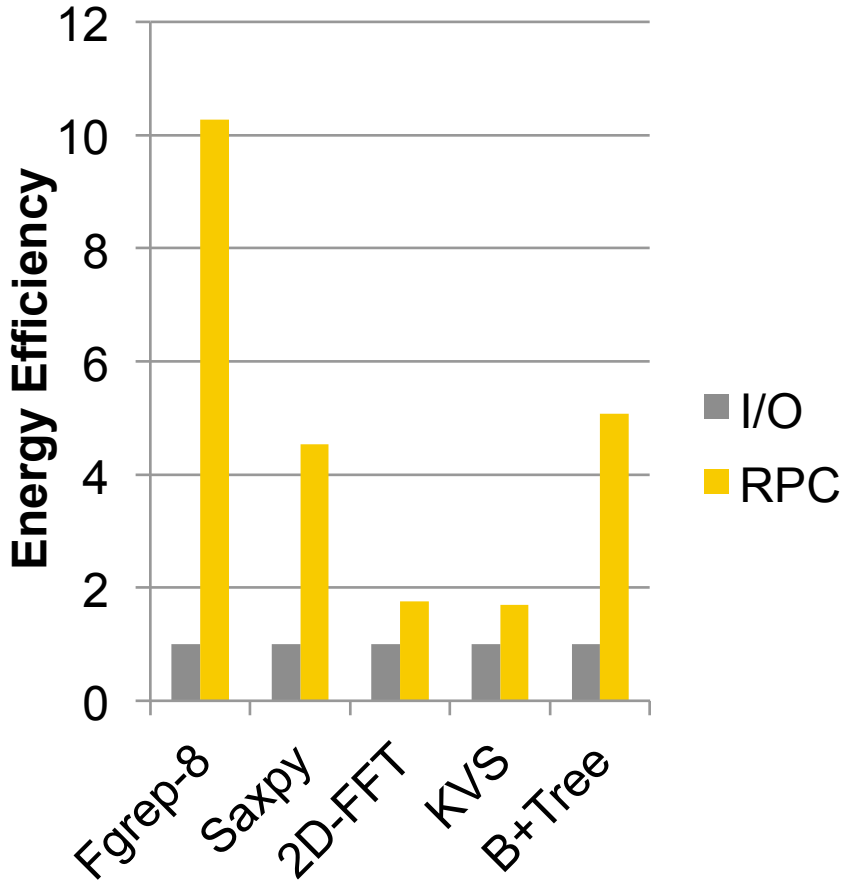
Run benchmarks and mini-apps in real-time

Estimate power draw and energy for different technology nodes

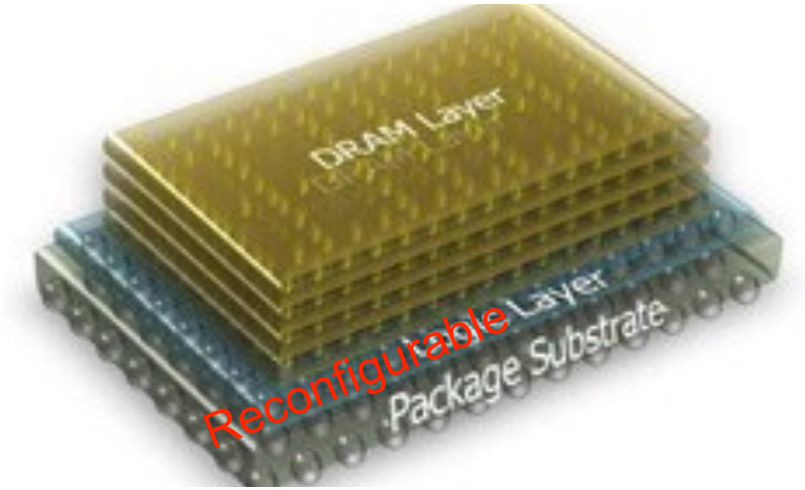
- Streaming string search
  - 8+× performance
  - 10+× energy efficiency
- Saxpy
  - 4× performance
  - 4.5× energy efficiency
- Key-Value store (key lookup)
  - 1.6× performance
  - 1.7× energy efficiency
- B+ tree search
  - 4+× performance
  - 5× energy efficiency

RESULTS


Speedup
10
9
8
7
6
5
4
3
2
1
0
Fgrep-8
Saxpy
2D-FFT
KVS
B+Tree
I/O
RPC

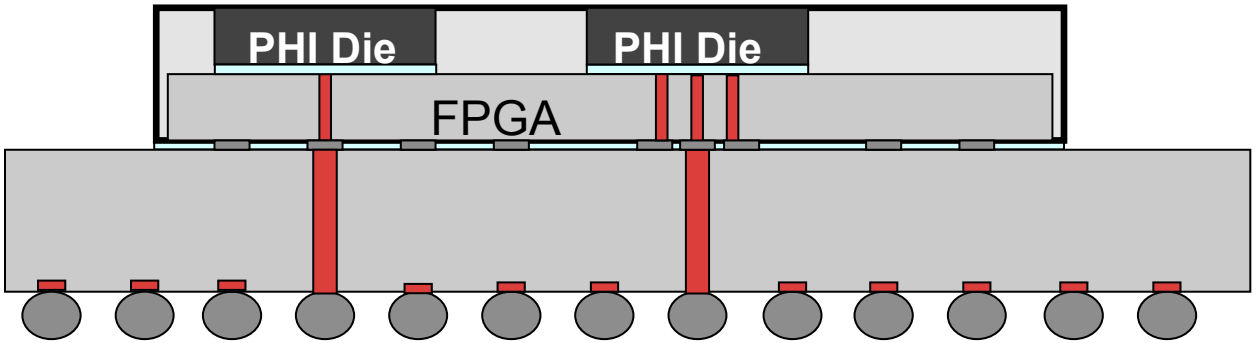

Energy Efficiency
12
10
8
6
4
2
0
Fgrep-8
Saxpy
2D-FFT
KVS
B+Tree
I/O
RPC

HOW ABOUT A RECONFIGURABLE LOGIC LAYER?

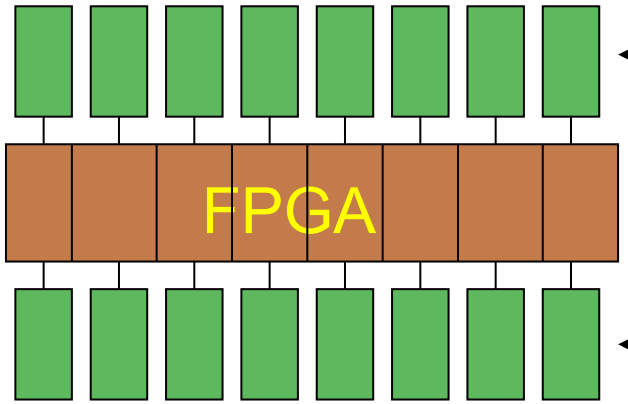


3D package

- DRAM stack
- TSVs directly into FPGA
- Has the time come for "Programmable Heterogeneous Interconnect"

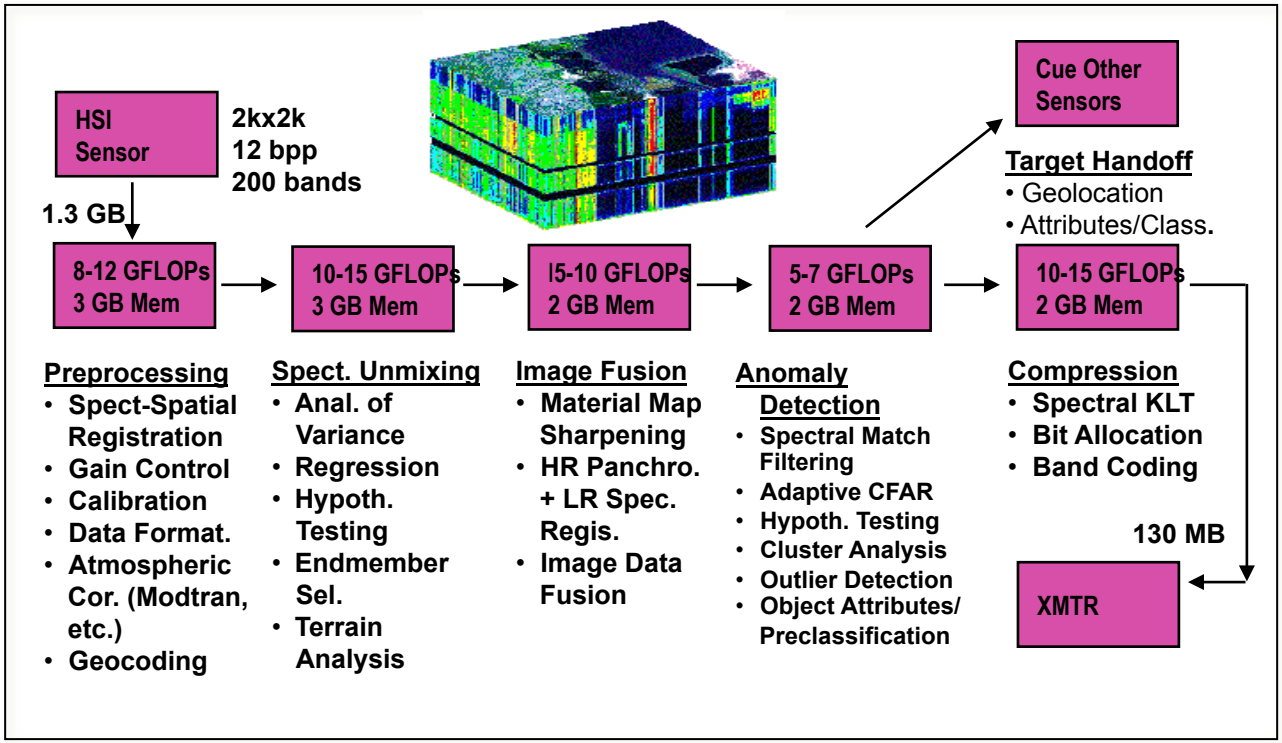


FPGA-BASED MEMORY INTENSIVE ARCHITECTURE



FPGA-resident compute engines operate directly on data in the 3D memory stack

Application driver: Processing hyperspectral imagery



XILINX IP

Methods and apparatus for implementing a stacked memory programmable integrated circuit system in package

US 7701251 B1

ABSTRACT

Methods and apparatus for implementing a stacked memory-programmable integrated circuit system-in-package are described. An aspect of the invention relates to a semiconductor device. A first integrated circuit (IC) die is provided having an array of tiles that form a programmable fabric of a programmable integrated circuit. A second IC die is stacked on the first IC die and connected therewith via inter-die connections. The second IC die includes banks of memory coupled to input/output (IO) data pins. The inter-die connections couple the IO data pins to the programmable fabric such that all of the banks of memory are accessible in parallel.

Arifur Rahman, Chidamber R. Kulkarni

Original Assignee Xilinx, Inc.

CONCLUSIONS

Near data processing is even more technically feasible than the 90's.

Commercial drivers may emerge to make it a reality in commodity DRAM

Reconfigurable logic offers opportunity to create custom near data applications at relatively low cost