A Sparse Matrix Personality for the Convey HC-1

Dept. of Computer Science and Engineering
University of South Carolina

Krishna K Nagar, Jason D. Bakos
Heterogeneous and Reconfigurable Computing Lab (HeRC)
http://herc.cse.sc.edu

This material is based upon work supported by the National Science Foundation under Grant Nos. CCF-0844951 and CCF-0915608.

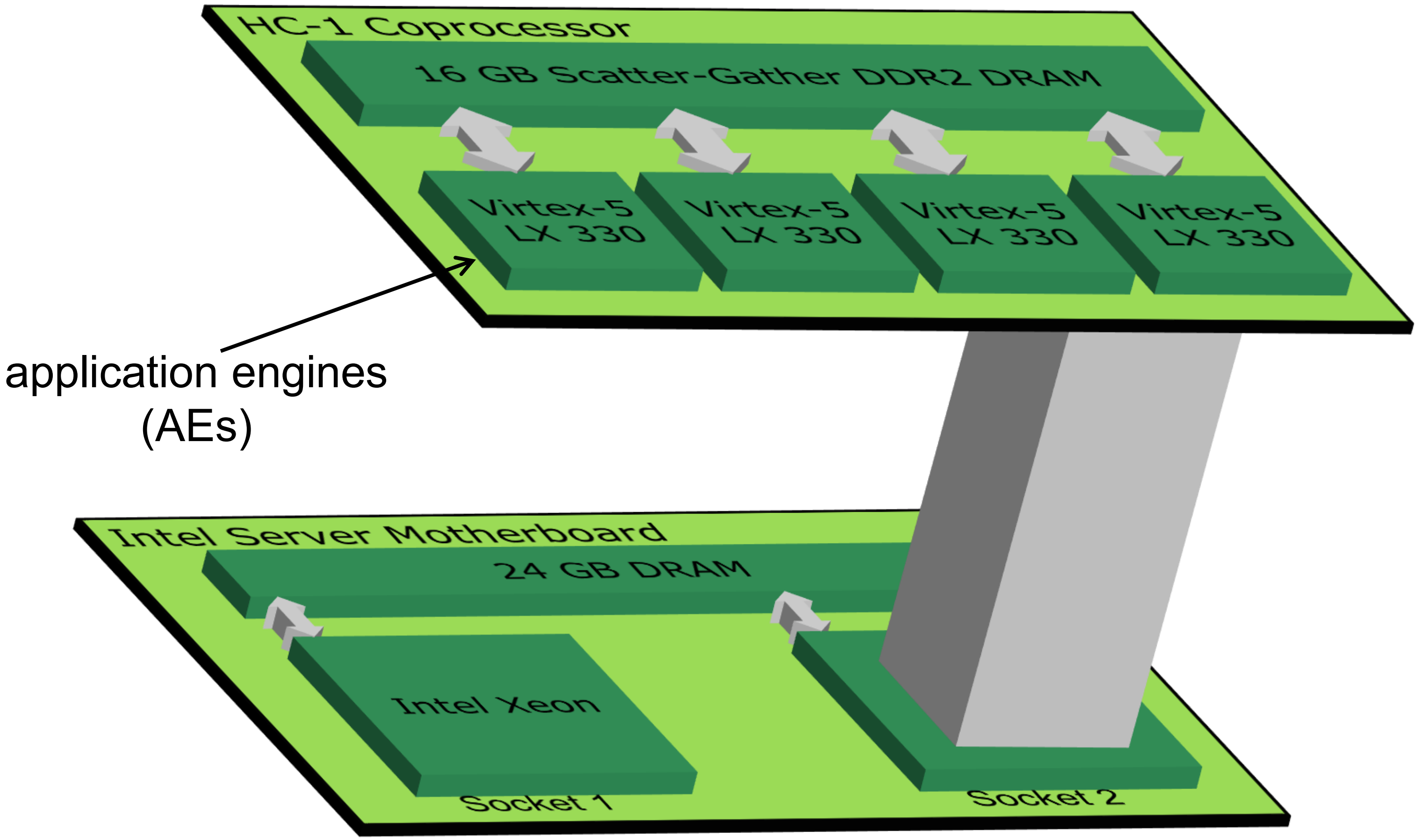
Introduction

- Convey HC-1: A turnkey reconfigurable computer
- Personality: A configuration of the user programmable FPGAs that works within the HC-1's execution and programming model
- This paper introduces a sparse matrix personality for Convey HC-1

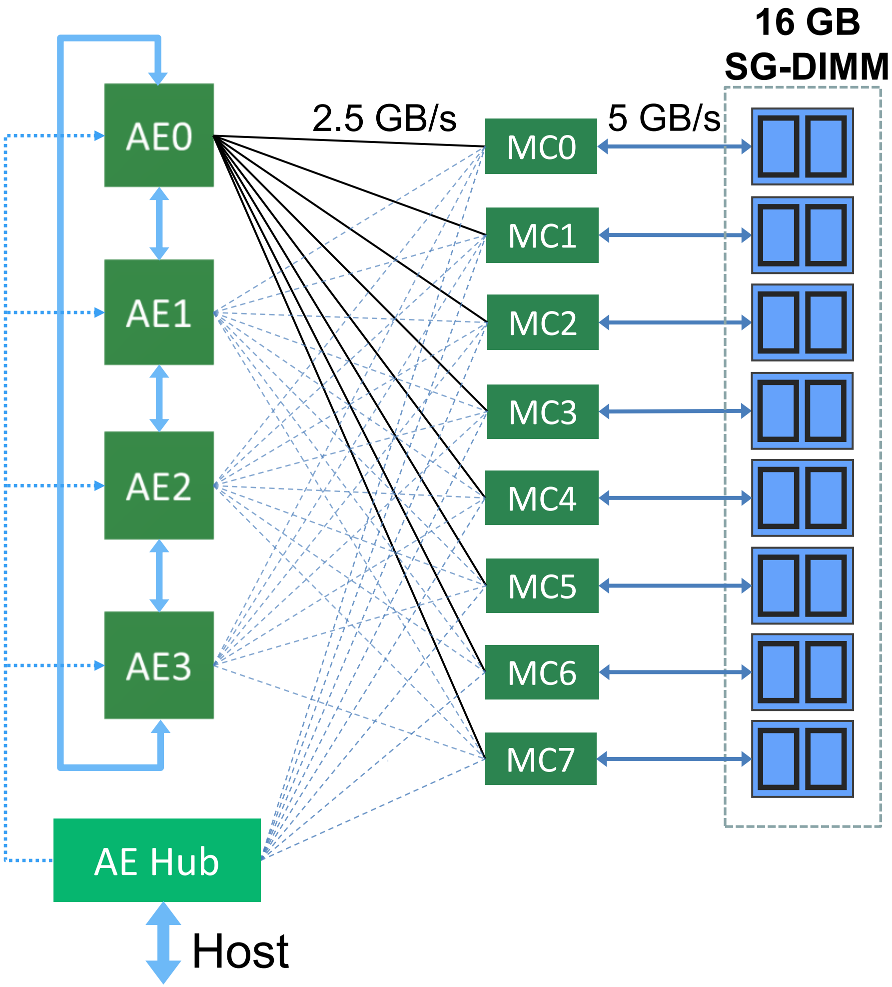
Outline

- Convey HC-1
  - Overview of system
  - Shared coherent memory model
  - High-performance coprocessor memory
- Personality design for sparse matrix vector multiply
  - Indirect addressing of vector data
  - Streaming double precision reduction architecture
- Results and comparison with NVIDIA Tesla

Convey HC-1


HC-1 Coprocessor
16 GB Scatter-Gather DDR2 DRAM
Virtex-5 LX 330
Virtex-5 LX 330
Virtex-5 LX 330
Virtex-5 LX 330
application engines (AEs)
Intel Server Motherboard
24 GB DRAM
Intel Xeon
Socket 1
Socket 2

Coprocessor Memory System



- Each AE connected to 8 MCs through a full crossbar
- Address space partitioned across all 16 DIMMs
- High-performance memory
  - Organized in 1024 banks
  - Crossbar parallelism gives 80 GB/s aggregate bandwidth
  - Relaxed memory model
- Smallest contiguous unit of data that can be read at full bandwidth = 512 BYTES

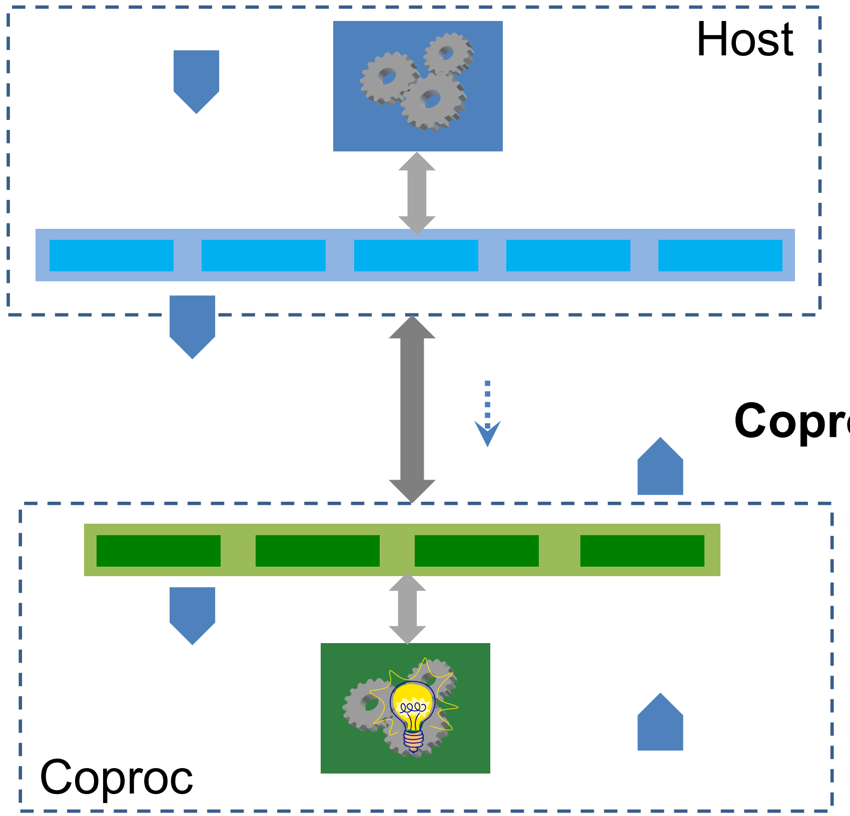
HC-1 Execution Model

Host Action

Invokes coprocessor;
Sends pointers to
memory blocks

Host Memory State

Input	Output
Shared	Shared
Exclusive	Invalid
Invalid	



Coprocessor Action

Read contents of
memory block

Write to a memory
block; Finish!

Coprocessor Memory State

Input	Output
Shared	Exclusive
Invalid	Shared
	Invalid

Convey Licensed Personalities

- Soft-core vector processor
  - Includes corresponding vectorizing compiler
  - Supports single and double precision
  - Supports hardware instructions for transcendental and random number generation
- Smith-Waterman sequence alignment personality
- No sparse matrix personality

Outline

- Convey HC-1
  - Overview of system
  - Shared coherent memory
  - High-performance coprocessor memory
- Personality design for sparse matrix vector multiply
  - Indirect addressing of vector data
  - Streaming double precision reduction architecture
- Results and comparison with NVIDIA CUSPARSE on Tesla and Fermi

Sparse Matrix Representation

- Sparse Matrices can be very large but contain few non-zero elements
- Compressed formats are often used, e.g.
  Compressed Sparse Row (CSR)

$$\begin{pmatrix} 1 & -1 & 0 & -3 & 0 \\ -2 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 6 & 4 \\ -4 & 0 & 2 & 7 & 0 \\ 0 & 8 & 0 & 0 & -5 \end{pmatrix}$$

val	(1	-1	-3	-2	5	4	6	4	-4	2	7	8	-5)
col	(0	1	3	0	1	2	3	4	0	2	3	1	4)
ptr	(0	3	5	8	11	13)							

Sparse Matrix-Vector Multiply

- Code for $Ax = b$

```
row = 0
for i = 0 to number_of_nonzero_elements do
   if i == ptr[row+1] then row=row+1, b[row]=0.0
  b[row] = b[row] + val[i] * x[col[i]]
end
```

recurrence (reduction)

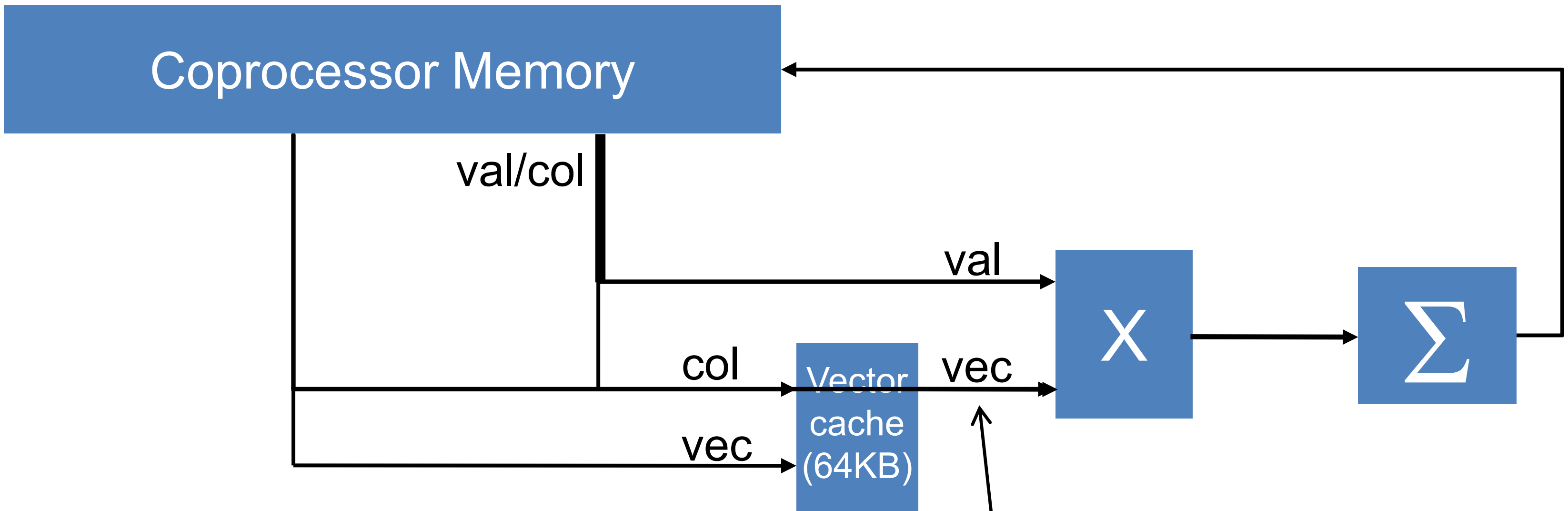
Low arithmetic intensity
(1 FLOP / 10 bytes)

indirect indexing

- NVIDIA GPUs achieve only 0.6% to 6% of their peak double precision performance with CSR SpMV

N. Bell, M. Garland, "Implementing Sparse Matrix-Vector Multiplication on Throughput-Oriented Processors," Proc. Supercomputing 2009.

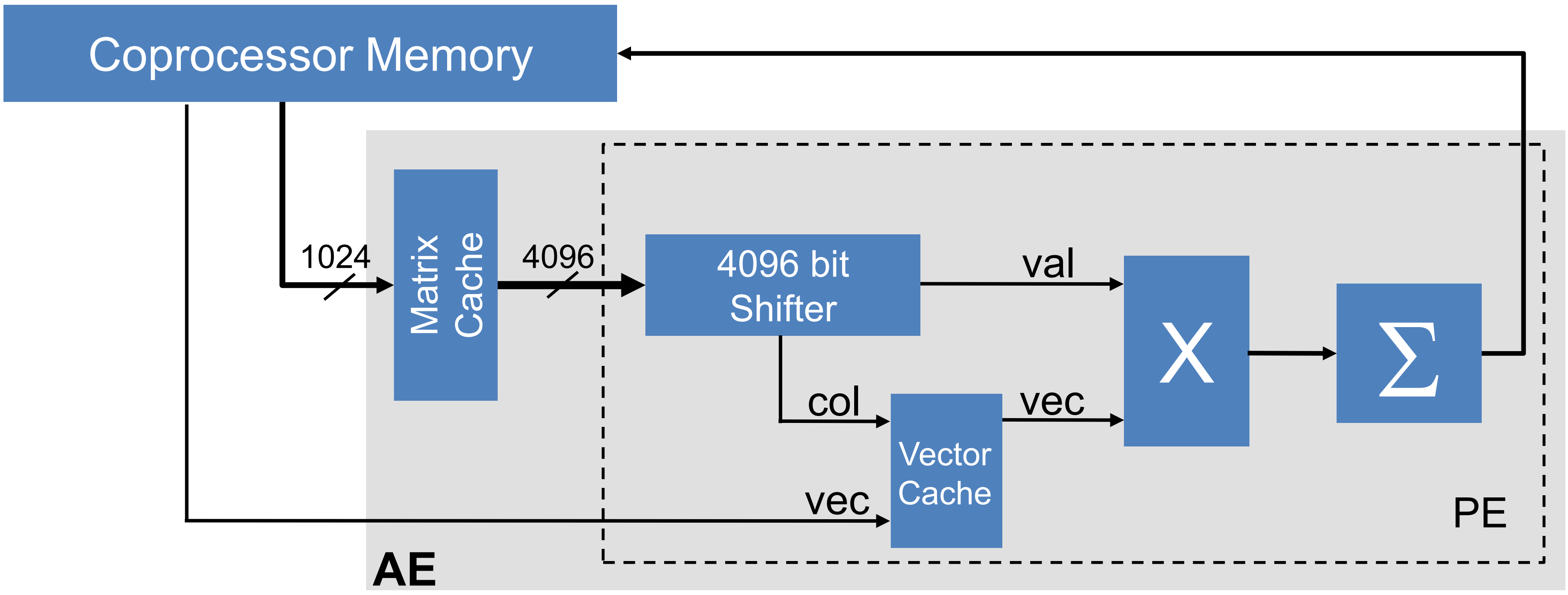
Indirect Addressing



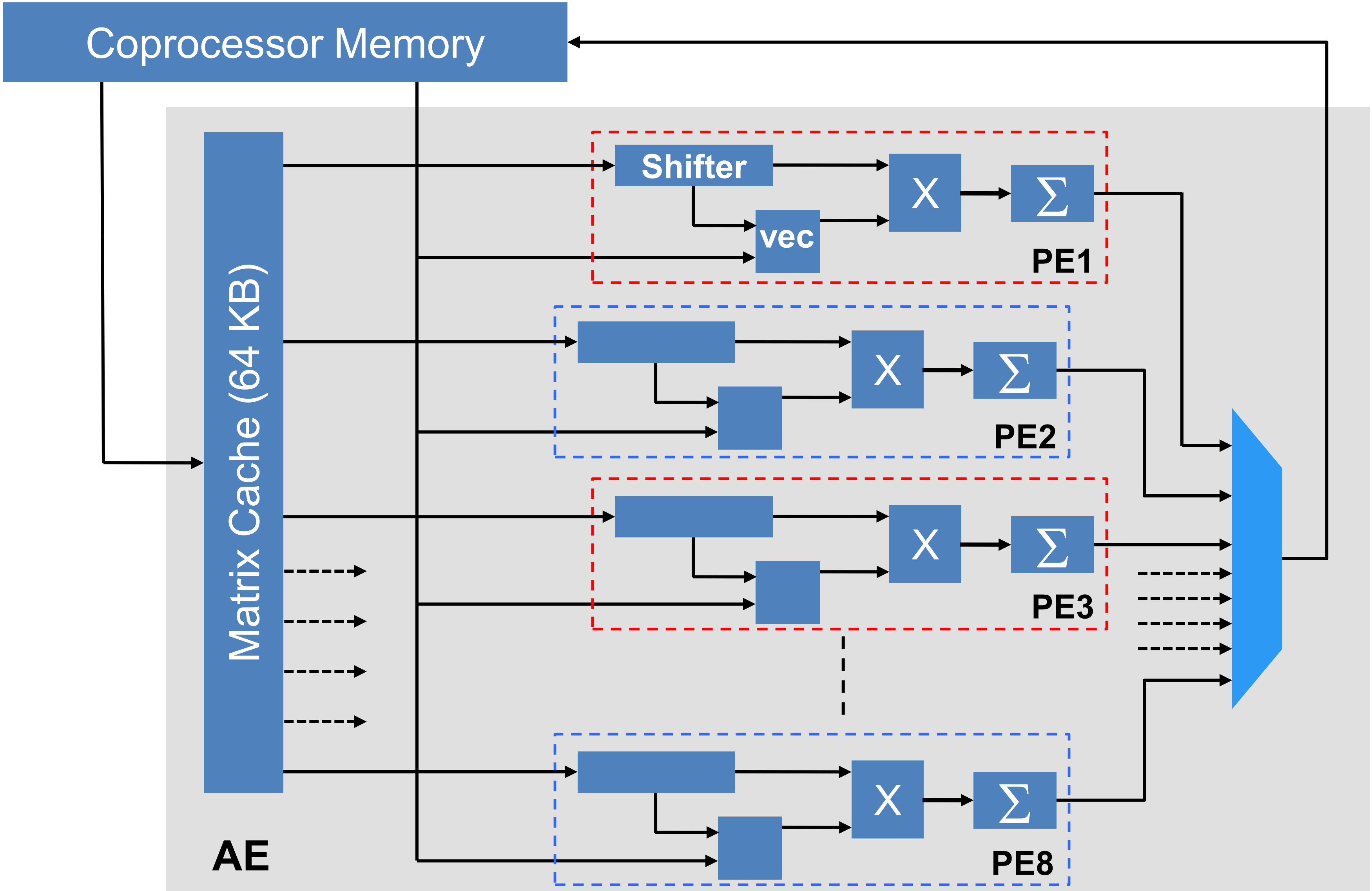

```
b[row] = b[row] + val[i] * x[col[i]]
```

Data Stream

- Matrix cache to get contiguous data
- Shifter loads matrix data in parallel and delivers serially to MAC



Top Level Design and Scaling

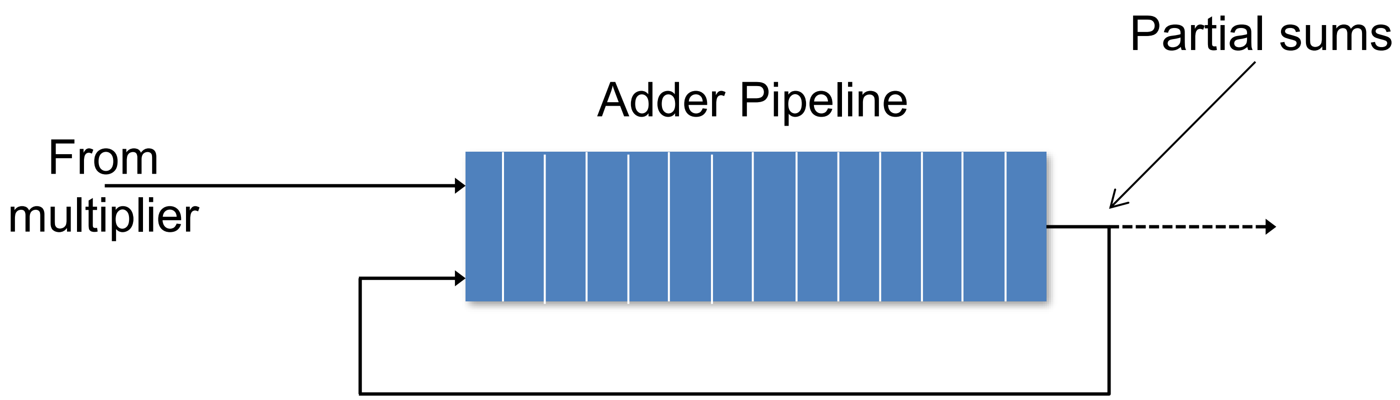


Outline

- Convey HC-1
  - Overview of system
  - Shared coherent memory
  - High-performance coprocessor memory
- Personality design for sparse matrix vector multiply
  - Indirect addressing of vector data
  - Streaming double precision reduction architecture
- Results and comparison with NVIDIA CUSPARSE on Tesla and Fermi

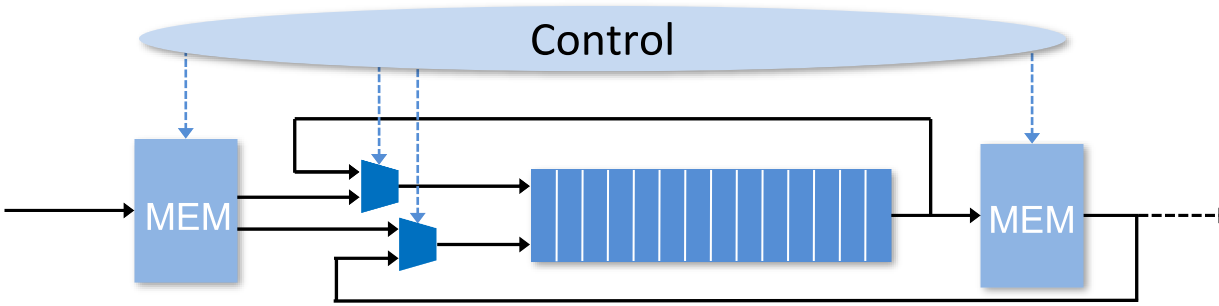
The Reduction Problem

- (Ideally) New values arrive every clock cycle
- Partial sums of different accumulation sets become intermixed in the deeply pipelined adder pipeline
  - Data hazard



Resolving Reduction Problem

- Custom architecture to dynamically schedule concurrent reduction operations

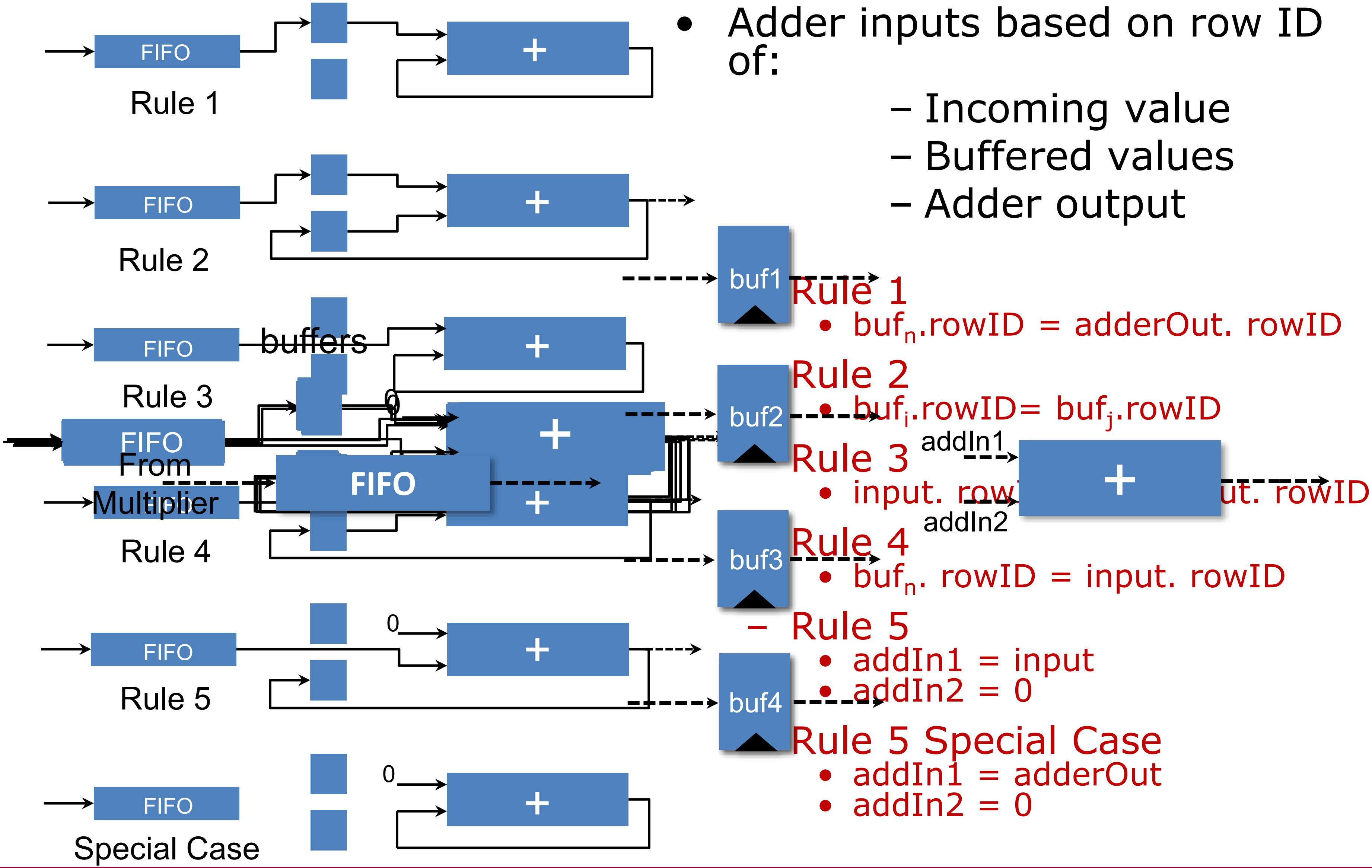


Group	Adders	Reduction BRAM
Prasanna '07	2	3
Prasanna '07	1	6
Gerards '08	1	9
This Work	1	3

Our Approach

- Built around 14 stage double precision adder
- Rule based approach
  - Governs the routing of incoming values and adder output
  - Decides inputs to the adder
  - Applied based on current state of the system
- Goal
  - Maximize the adder utilization
  - Minimize the required number of buffers
- Used software model to design rules and find required number of buffers

Reduction Circuit



- Adder inputs based on row ID of:
  - Incoming value
  - Buffered values
  - Adder output

Rule 1
- buf_n.rowID = adderOut. rowID

Rule 2
- buf_i.rowID= buf_j.rowID

Rule 3
- input. rowID = adderOut. rowID

Rule 4
- buf_n. rowID = input. rowID

– Rule 5
- addIn1 = input
- addIn2 = 0

Rule 5 Special Case
- addIn1 = adderOut
- addIn2 = 0

Outline

- Convey HC-1
  - Overview of system
  - Shared coherent memory
  - High-performance coprocessor memory
- Personality design for sparse matrix vector multiply
  - Indirect addressing of vector data
  - Streaming double precision reduction architecture
- Results and comparison with NVIDIA CUSPARSE on Tesla

SpMV on GPU

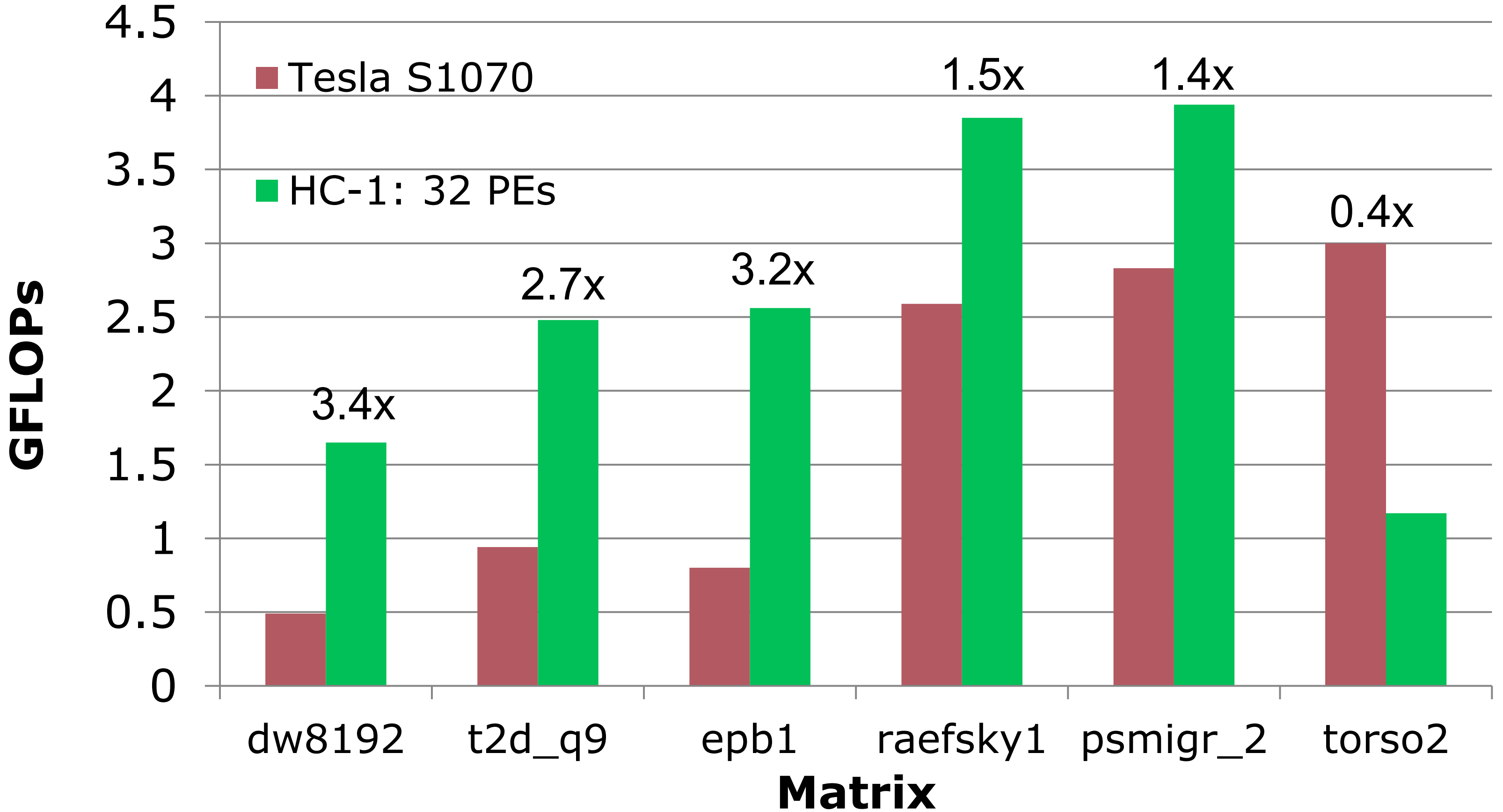
- GPUs widely used for accelerating scientific applications
- GPUs generally have more mem bandwidth than FPGAs, so do better for computations with low arithmetic intensity
- Target: NVIDIA Tesla S1070
  - Contains four Tesla T10 GPUs
  - Each GPU has 50% more memory bandwidth than all 4 AEs on Convey HC-1 *combined*
- Implementation using NVIDIA CUDA CUSPARSE library
  - Supports Sparse BLAS routines for various sparse matrix representations including CSR
  - Can run only on single GPU for single SpMV computations

Experimental Results

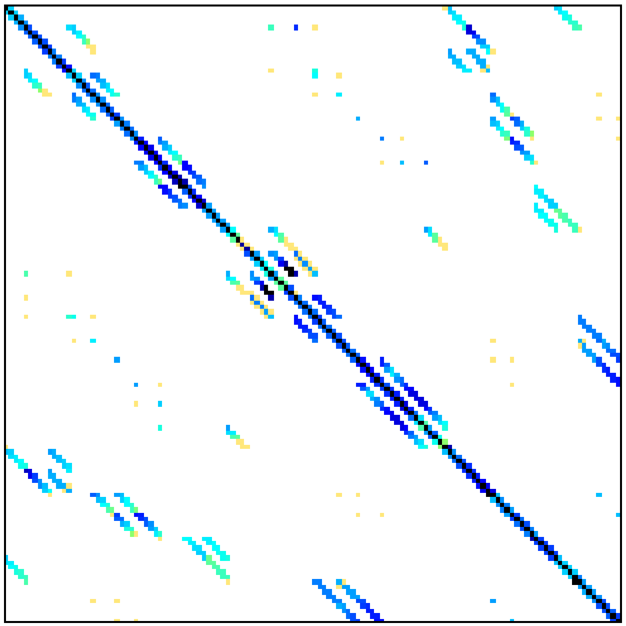
- Test matrices from Matrix Market and UFL Matrix collection
- Throughput = 2 * nz / (Execution Time)

Matrix	Application	r * c	nz	nz/row
dw8192	Electromagnetics	8192*8192	41746	5.10
t2d_q9	Structural	9801*9801	87025	8.88
epb1	Thermal	14734*14734	95053	6.45
raefsky1	Computational fluid dynamics	3242*3242	294276	90.77
psmigr_2	Economics	3140*3140	540022	171.98
torso2	2D model of a torso	115967*115967	1033473	8.91

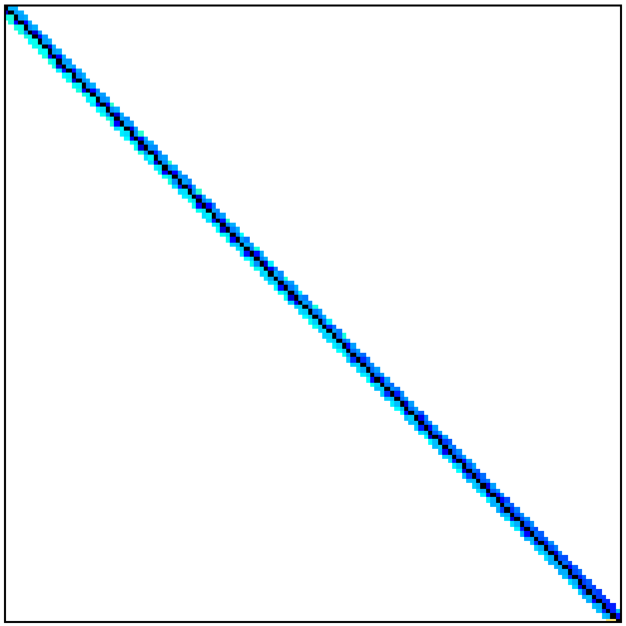
Performance Comparison



Test Matrices



torso2



epb1

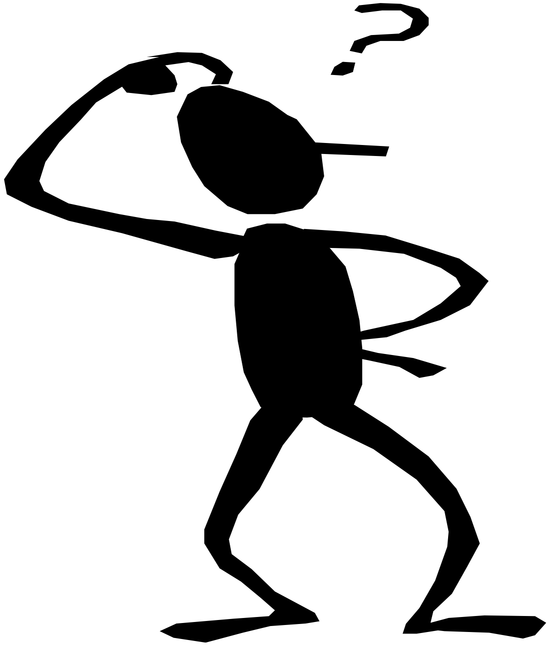
Final Word

- Conclusions
  - Described a SpMV personality tailored for Convey HC-1 built around new streaming reduction circuit architecture
  - FPGA outperforms GPU
  - Custom architectures have the potential for achieving high performance for kernels with low arithmetic intensity
- Future Work
  - Analyze design tradeoffs between vector cache and functional units
  - Improve the vector cache performance
  - Multi-GPU implementation

About Us

Heterogeneous and Reconfigurable Computing Lab
The University of South Carolina

Visit us at http://herc.cse.sc.edu

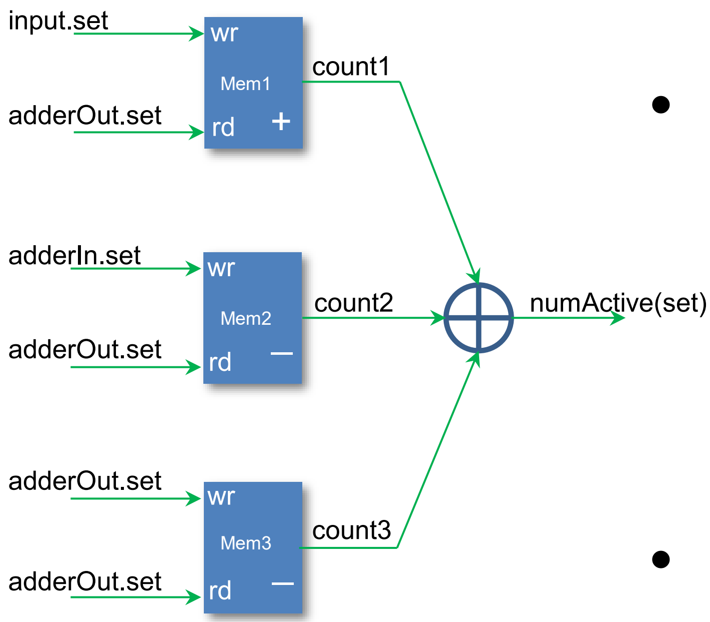


Thank You!

Resource Utilization

PEs	Slices	BRAM	DSP48E
4 per AE (Overall 16)	26055 / 51840 (50%)	146 / 288 (50%)	48 / 192 (25%)
8 per AE (Overall 32)	38225 / 51840 (73%)	210 / 288 (73%)	96 / 192 (50%)

Set ID Tracking Mechanism



- Three dual ported memories with respective counters
- Write Port
  - Counter1 always increments associated incoming value setID
  - Counter2 always decrements associated adder input setID
  - Counter 3 decrements when number of associated active values reach one setID
- Read Port
  - Outputs current value for associated setID
- Set is completely reduced and output when count1 + count2 + count3 = 1