

Processor in Memory Cortical Extensions

Paul Franzon

Department of Electrical and Computer Engineering
NC State University
919.515.7351, paulf@ncsu.edu

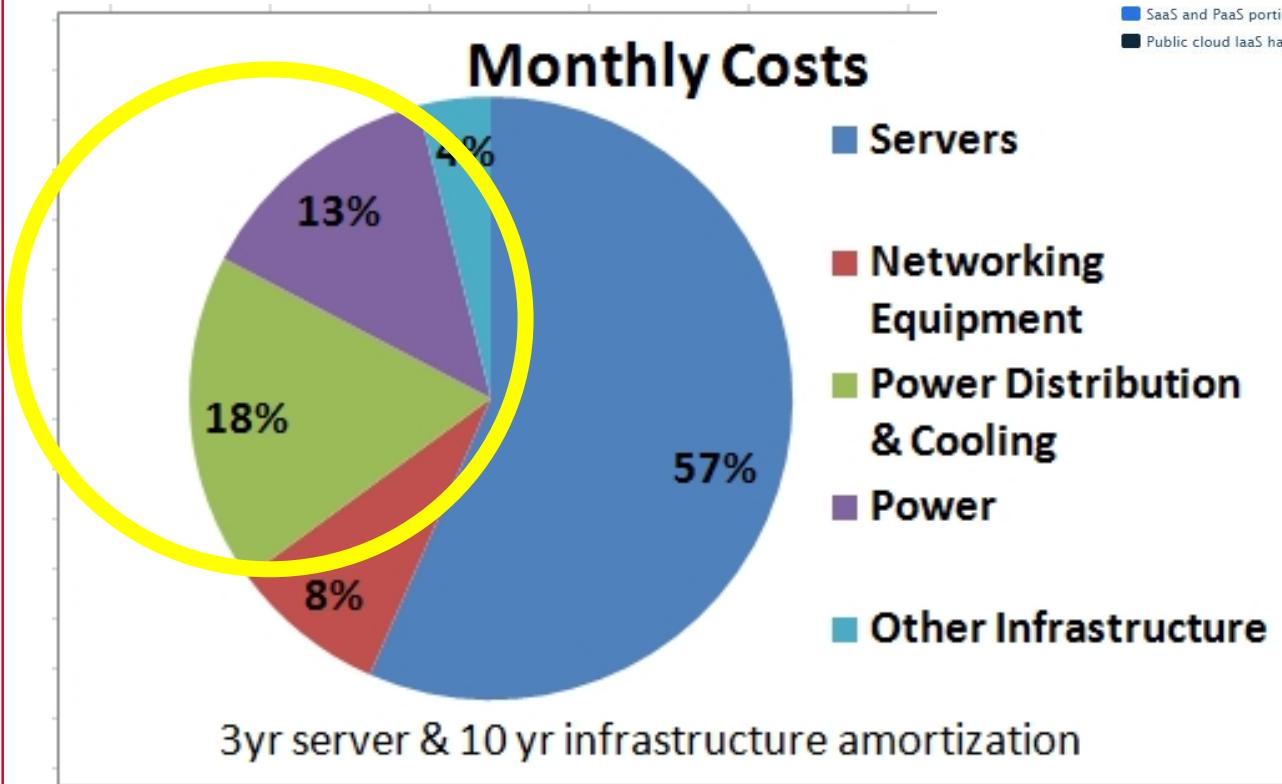
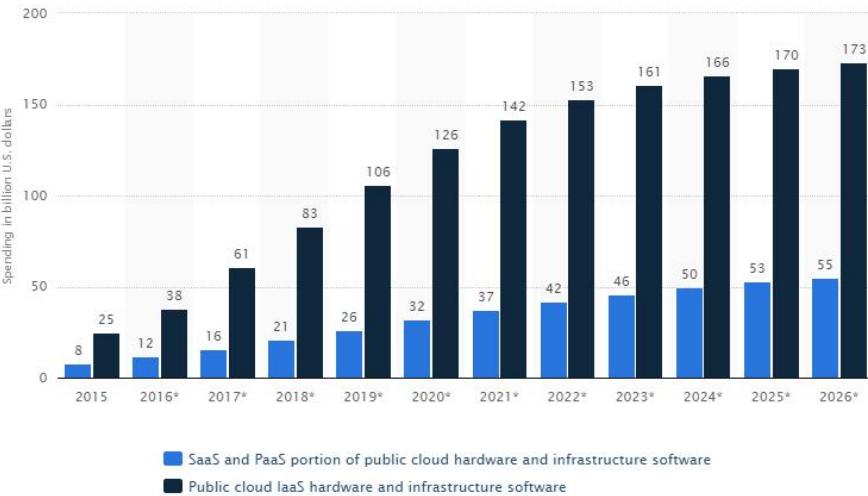
Team members: Lee Baker, Sumon Dey,
Josh Schabel, Weifu Li

Funding:

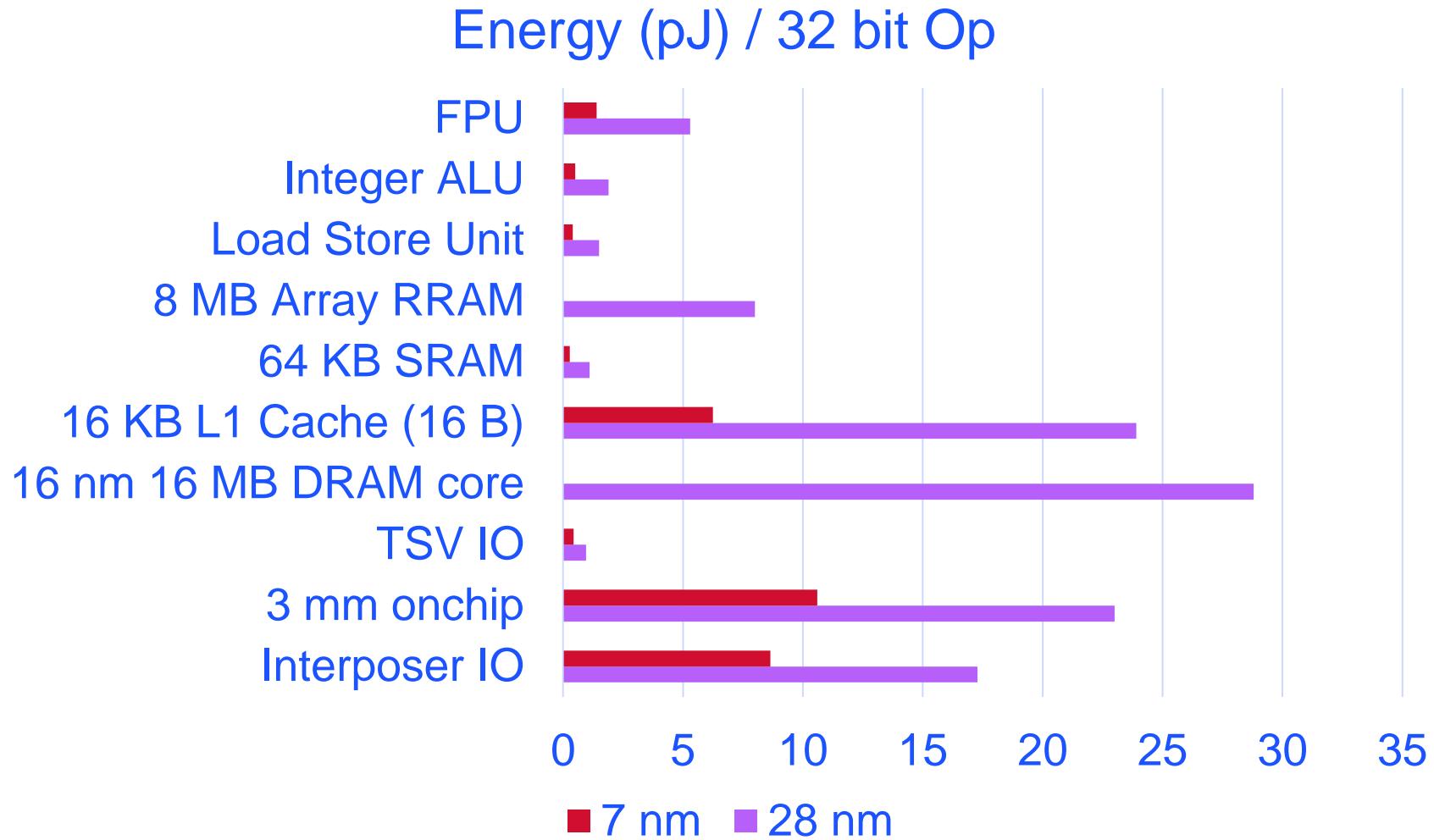


Server Power Costs

► ~\$20B/year worldwide

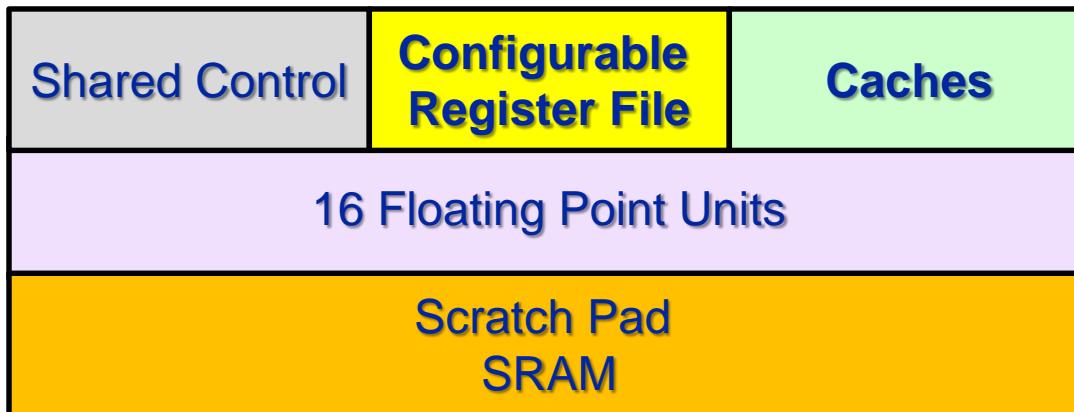


Energy/32-bit Op

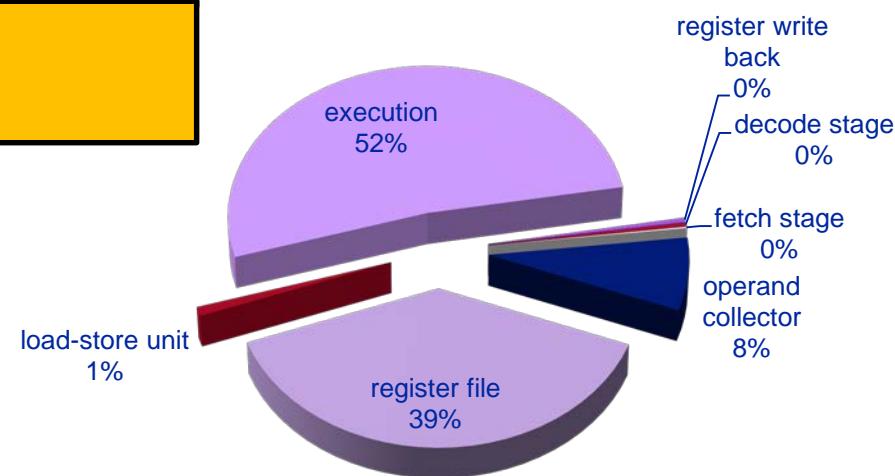


SIMD Compute Tile

► Single Instruction Multiple Data



Total power



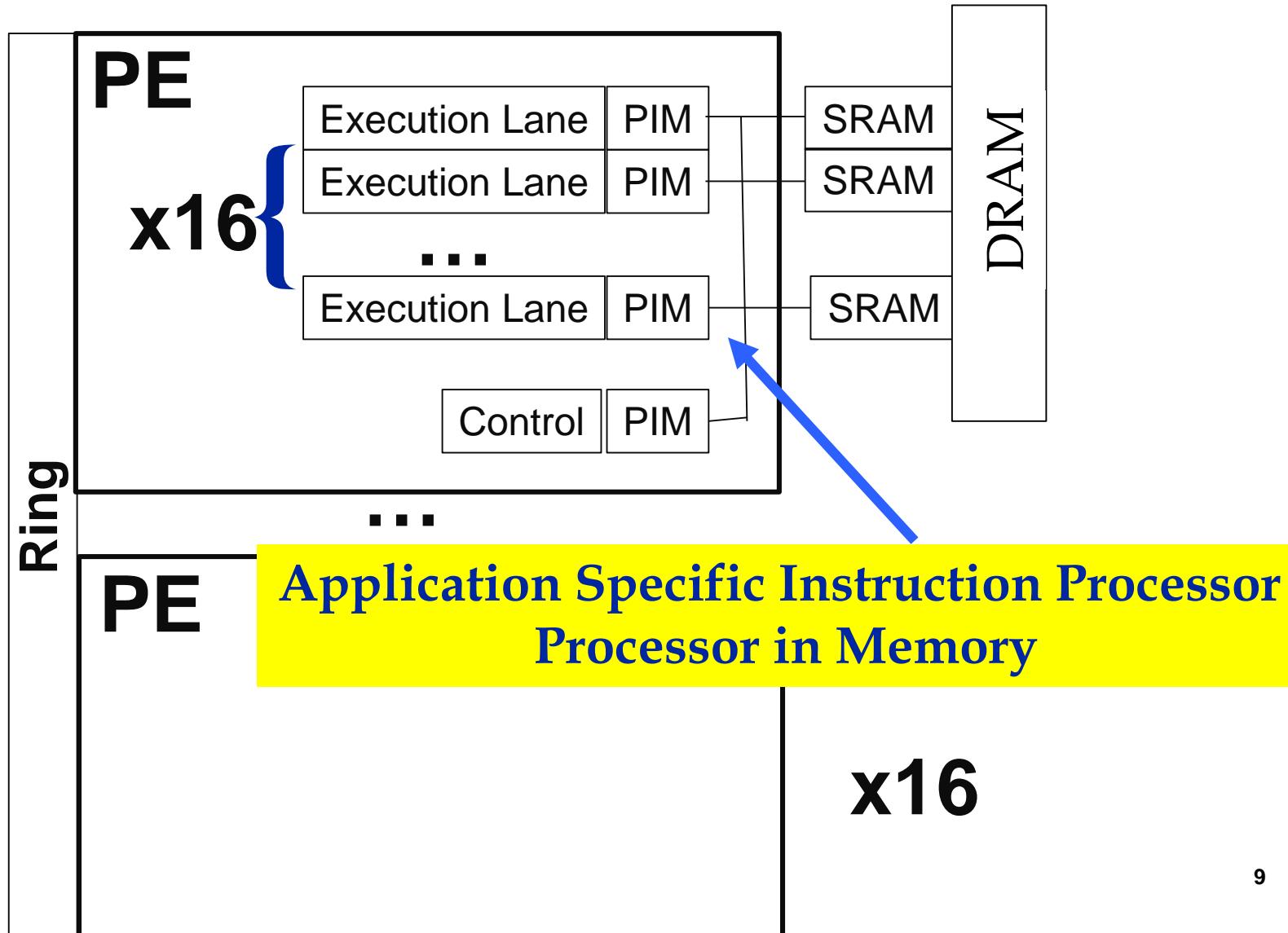
32 GFLOPS/W in 65 nm on FFT Benchmark

Cortical / Machine Learning Algorithms

Algorithm	Success
Convolutional Networks	Image & speech recognition; Under intense application driven exploration
Hierarchical Temporal Memory	Anomaly detection; “Digital” pattern matching
Cogent Confabulation	Credit Card Fraud Detection; Text Recognition
Sparsey	Video action recognition
LSTM	Incremental learning
LSM	Spiking network with STDP

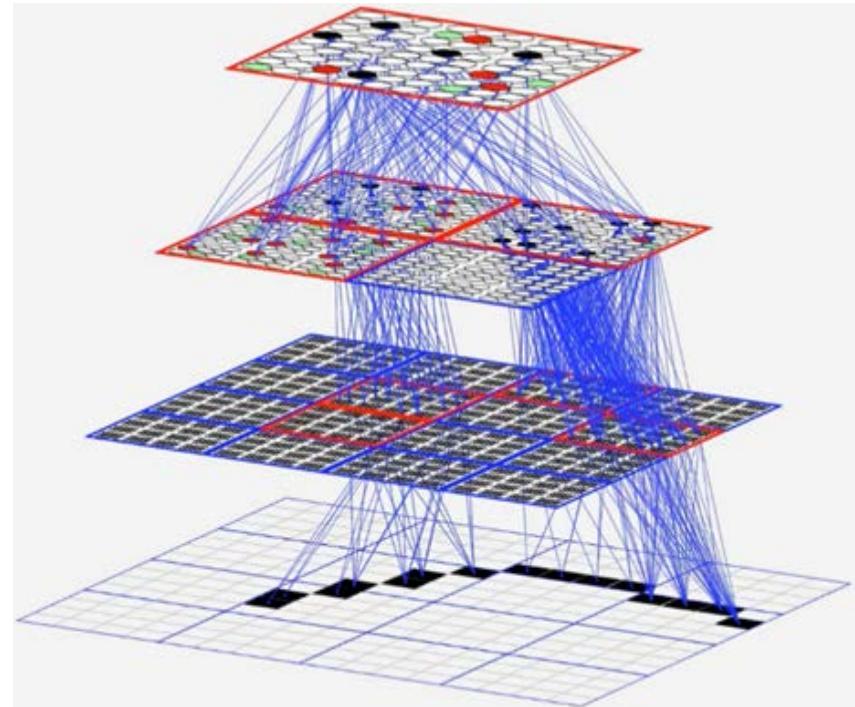
- ▷ These algorithms are interconnect & memory bound – speedup can be achieved in appropriate distributed architecture

ASIP with PIM



Sparsey

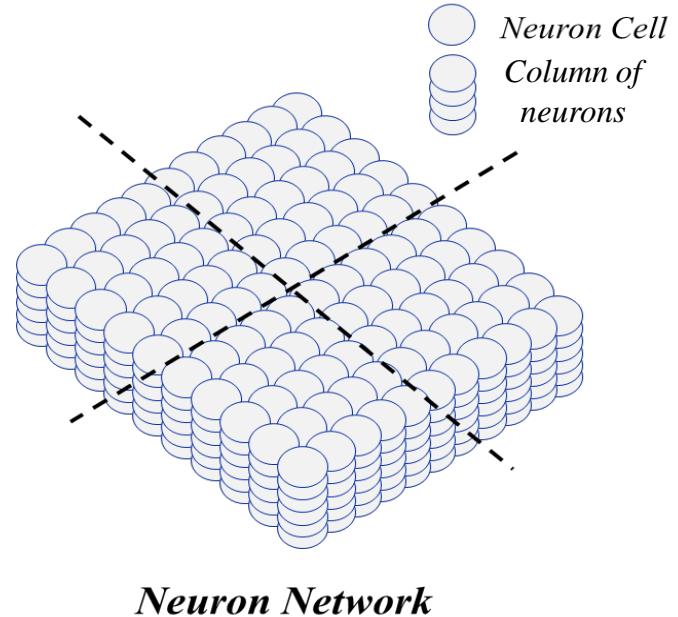
- ◆ Hierarchical
- ◆ Recurrent
- ◆ Spatial & temporal associativity
- ◆ Sparse distributed codes
- ◆ “One hot” learning
- ◆ Static, on/off synaptic connections
- ◆ Sigmoid neuron model
- ◆ **Highly parallelizable binary and floating point arithmetic**
- ◆ Neurons in competitive modules (CMs) compete to be active



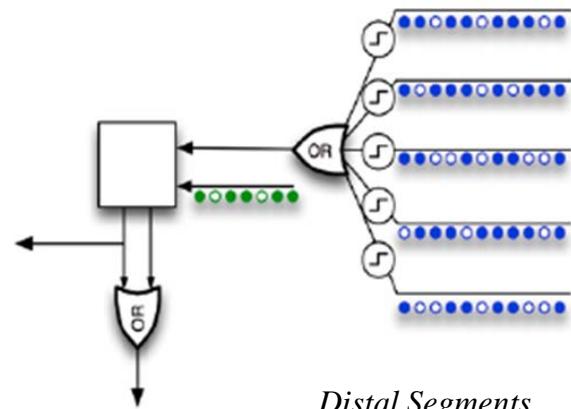
G.J. Rinkus, "A cortical sparses distributed coding model linking mini- and macro-column-scale functionality," in *Frontiers in Neuroanatomy* June 2010.

HTM

- ◆ Hierarchical
- ◆ Recurrent
- ◆ Sparse distributed representations
- ◆ “One hot” learning
- ◆ Statically connected synapses for spatial learning and inference
- ◆ Dynamically connected synapses for temporal learning, inference, and predictions
- ◆ Predictions lead to stability
- ◆ **Highly divergent binary and integer operations**

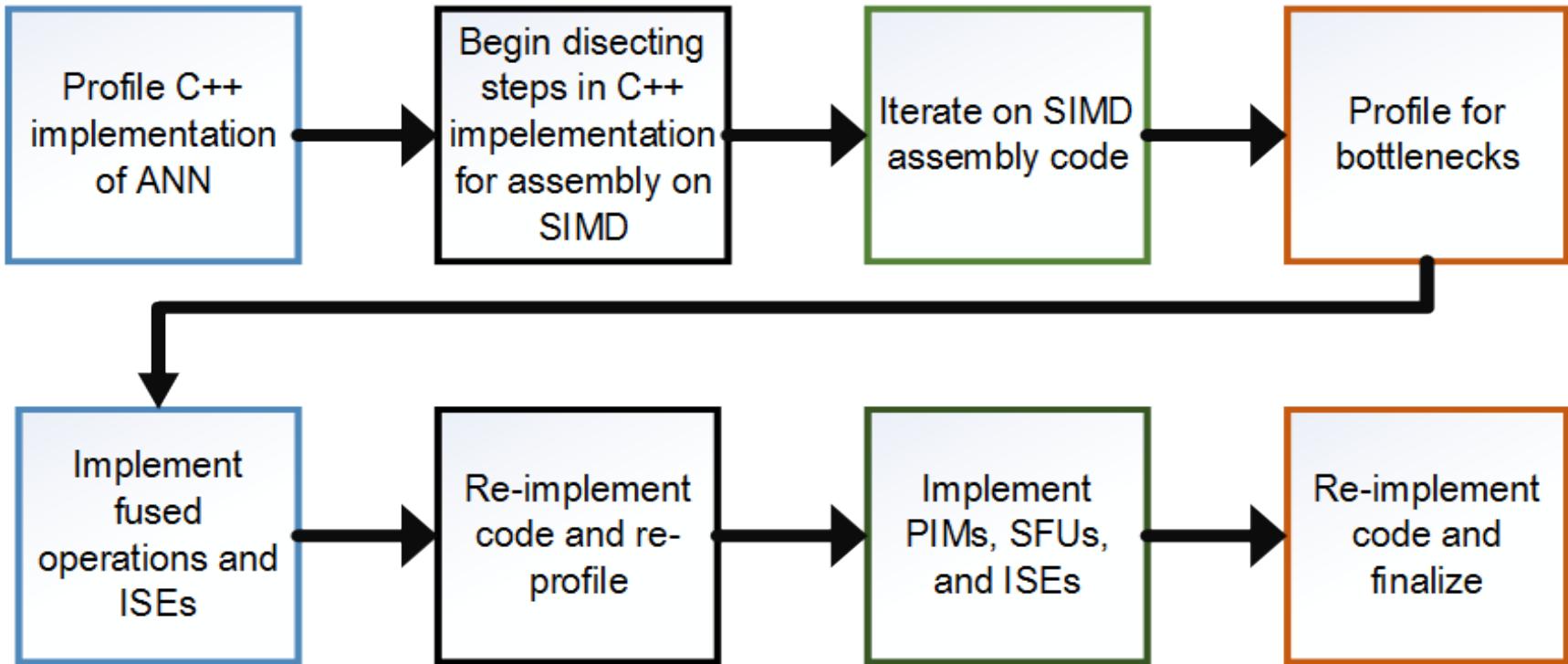


Neuron Network



Distal Segments

Technical Approach: CUDA to ASIP



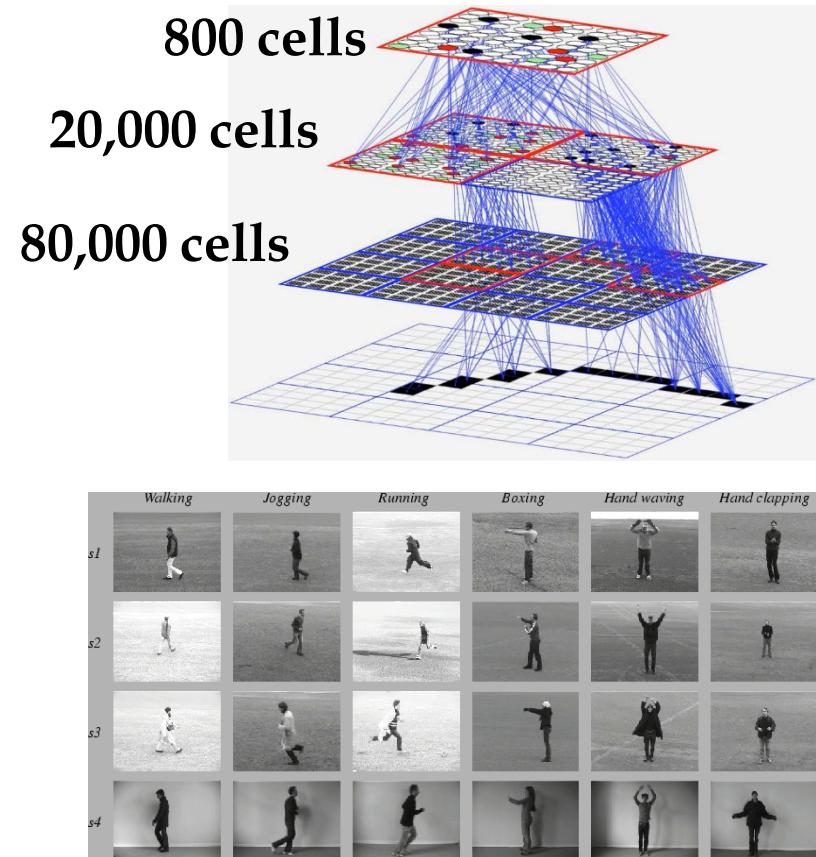
- ▶ Fully manual identification process of bottlenecks, ISEs, and PIMs
- ▶ Could use automated ISE identification as in:
 - ◆ Deepti Shrimal and Manoj Kumar Jalin, "Instruction Customization: A challenge in ASIP realization," IJAC, 2014.
 - ◆ D. Shapiro et al., "Asips for artificial neural networks," SACI, 2011.

GPU Implementation of Sparsey

► “Server class” GPU

- ◆ nVidia Tesla K20C with 2496 CUDA cores
- ◆ KTH Video “action recognition” benchmark

Parameter	Result	
Run time	8 ms / frame	
Power*	~210 W*	
Hot spots	Weight summation	80%
	Softmax	10%
Synapse Memory	~50 MB	



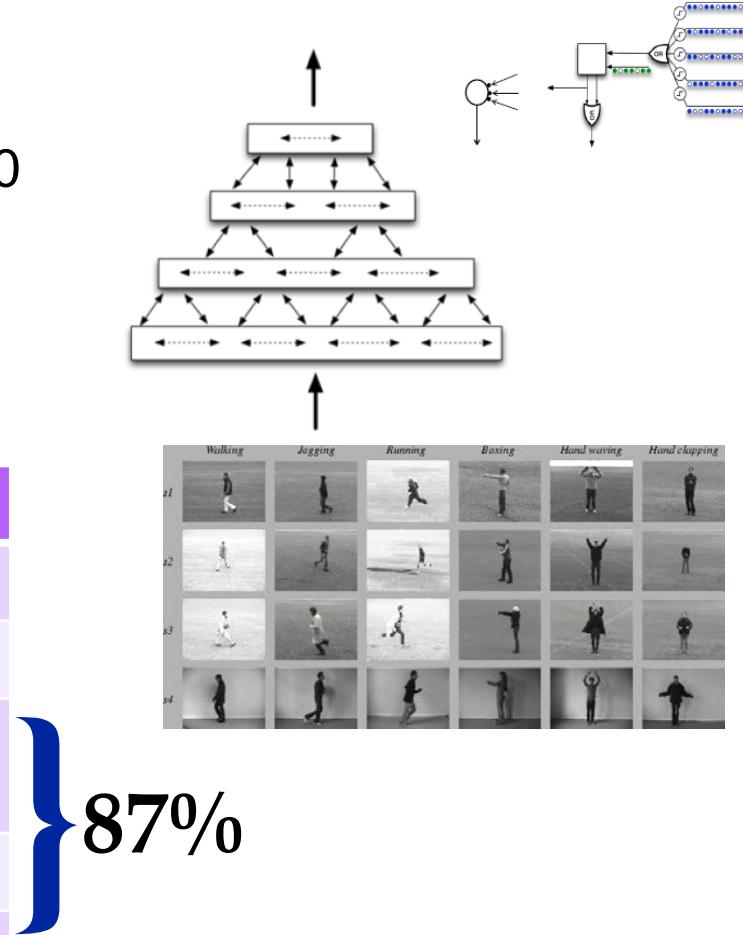
* datasheet

GPU Implementation of Numenta HTM

► “Consumer class” GPU

- ◆ 392 core Nvidia GeForce GT 640
- ◆ KTH Video “action recognition” benchmark

Parameter	Result	
Run time	225 ms / frame	
Power	~70 W*	
Hot spots	Get Column Overlap	33%
	ColumnInhibit	10%
Memory	CreateSegment	44%
	~100 MB	



* Datasheet

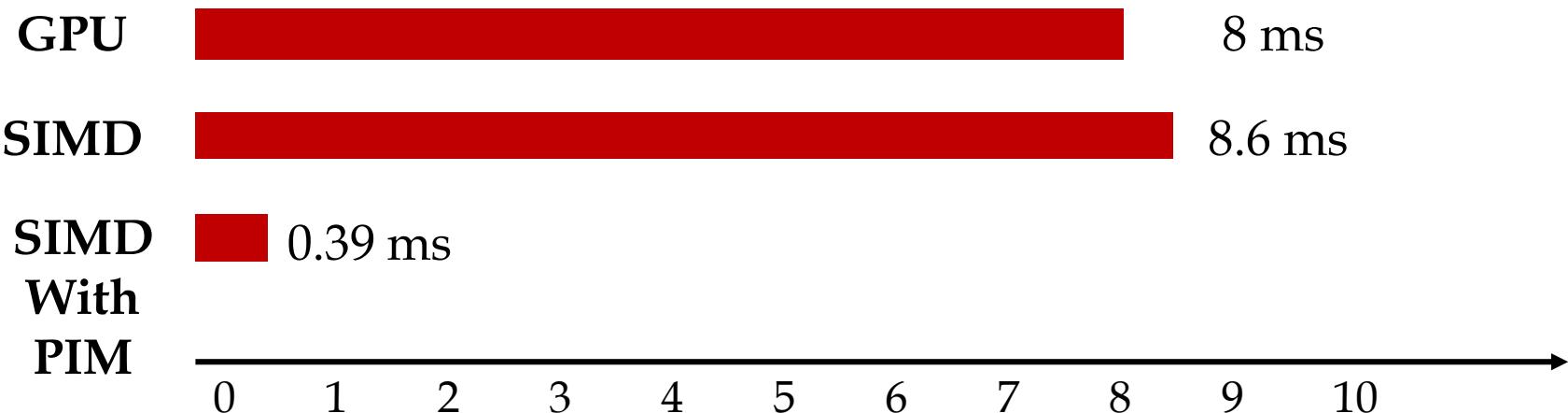
Cognitive ISA Extensions

- ▷ **PIM = Processor in Memory**

Sparsey	HTM
BSUM	F2I
	IMAX, IMIN
I2F	GETBIT
FMAX, FMIN	ICOMP (GT,LT,EQ,GTE,LTE)
ONEHOT	PIM-NODE-BITONIC-SORT-FP
PIM-SELECT-WINNER-LD	PIM-NODE-BITONIC-SORT-INT
PIM-SELECT-WINNER-LE	PIM-CLUSTER-MULTI-MERGE
PIM-UPDATE-WINNER	PIM-GALAXY-MULTI-MERGE
PIM-WEIGHT-SUM	LDI or STI to address zero is null

Processor In Memory

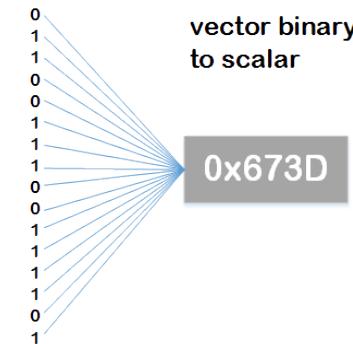
- ▷ **Accelerates key distributed tasks in both algorithms**
 - ◆ E.g. Sparsey weight summation implemented as PIM
 - ◆ Learning disabled results below (per frame of video)
 - ◆ Enabling learning adds about 15% to run time



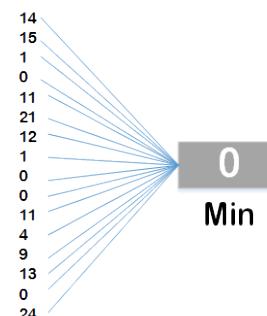
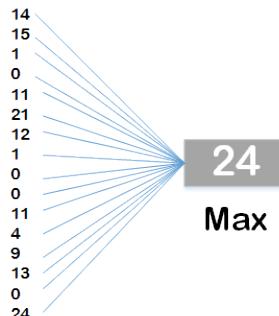
HTM Instruction Set Extensions

- ▶ Vector-to-scalar ops
- ▶ Binary-to-scalar ops

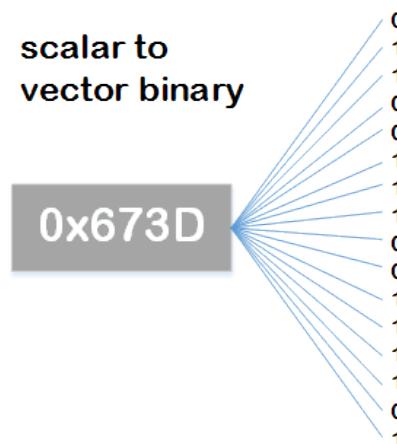
14
15 + 29 + 30
1 + 1 +
0 + 11 + 75
11 + 21 + 32 + 45
21 + 12 + 13 + 136
12 + 1 + 0 +
0 + 0 + 15 +
0 + 11 + 15 + 61
11 + 4 + 9 +
4 + 13 + 22 + 46
9 + 0 + 13 +
13 + 0 + 24 + 24
0 + 24 +
24 + 24 +
Summation



- ▶ Min, Max



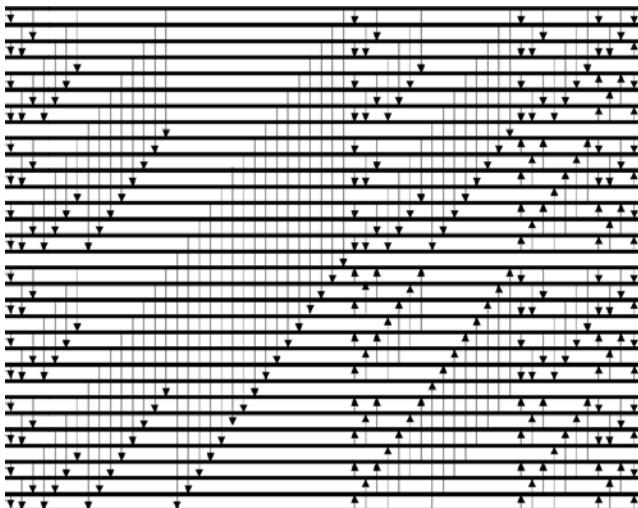
- ▶ Scalar-to-binary ops



HTM PIMs

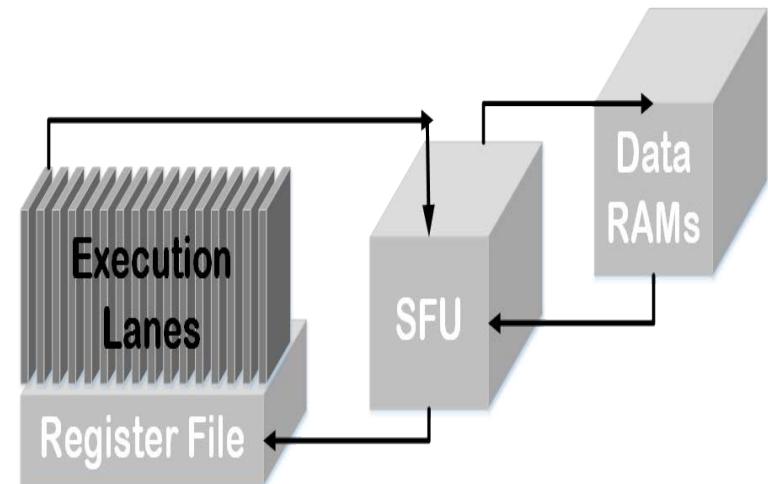
► Bitonic Sorter

- ◆ 64 element integer sorter for sorting column IDs in spatial pooler
- ◆ 32 element FP sorter for sorting Euclidean distances in temporal pooler



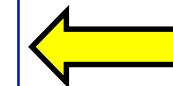
► Scatter-gather

- ◆ Temporal pooling phase, creating new distal synapses
- ◆ Streams data from RAM like PIM
- ◆ Works on data from execution lanes' RF port

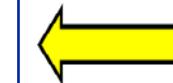


Results @ 65 nm, 250 MHz

	GPU Baseline	32 16-wide SIMD	32 16-wide SIMD w/ PIMs
Tech. node	28 nm	65 nm	65 nm
Sparsey			
Latency per Frame	8 ms	9.26 ms	526 µs
Power	200 W*	2.025 W	2.025 W***
Logic Area	561 mm ²	25.5 mm ²	26.7 mm ²
Performance per Power	1	85	1490
Performance per Area	1	19	319
HTM			
Latency per Frame	225 ms	100 ms	12.7 ms
Power	70 W**	2.308 W	2.308 W***
Logic Area	118 mm ²	25.5 mm ²	29.7 mm ²
Performance per Power	1	68	537
Performance per Area	1	10	70



~16x
Speed up!!!



~17x
Speed up!!!

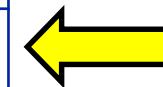
*Tesla K20C datasheet, 706 MHz

**GeForce GT 640 datasheet, 902 MHz

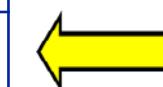
***Does not compensate for when SIMD is stalled and PIMs are on. Just SIMD power.

Results @ 65 nm, 250 MHz

	GPU Baseline	32 16-wide SIMD	32 16-wide SIMD w/ PIMs
Tech. node	28 nm	65 nm	65 nm
Sparsey			
Latency per Frame	8 ms	9.26 ms	526 µs
Power	200 W*	2.025 W	2.025 W***
Logic Area	561 mm ²	25.5 mm ²	26.7 mm ²
Performance per Power	1	85	1490
Performance per Area	1	19	319
HTM			
Latency per Frame	225 ms	100 ms	12.7 ms
Power	70 W**	2.308 W	2.308 W***
Logic Area	118 mm ²	25.5 mm ²	29.7 mm ²
Performance per Power	1	68	537
Performance per Area	1	10	70



**1490x
power
efficient**

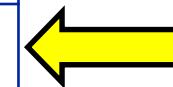


**~537x
Power
efficient**

Results @ 65 nm, 250 MHz

	GPU Baseline	32 16-wide SIMD	32 16-wide SIMD w/ PIMs
Tech. node	28 nm	65 nm	65 nm
Sparsey			
Latency per Frame	8 ms	9.26 ms	526 µs
Power	200 W*	2.025 W	2.025 W***
Logic Area	561 mm ²	25.5 mm ²	26.7 mm ²
Performance per Power	1	85	1490
Performance per Area	1	19	319
HTM			
Latency per Frame	225 ms	100 ms	12.7 ms
Power	70 W**	2.308 W	2.308 W***
Logic Area	118 mm ²	25.5 mm ²	29.7 mm ²
Performance per Power	1	68	537
Performance per Area	1	10	70

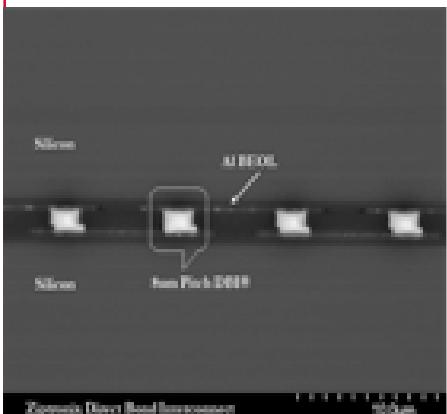
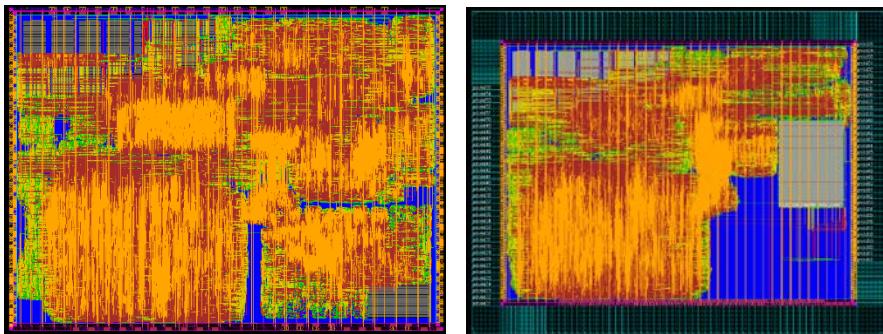
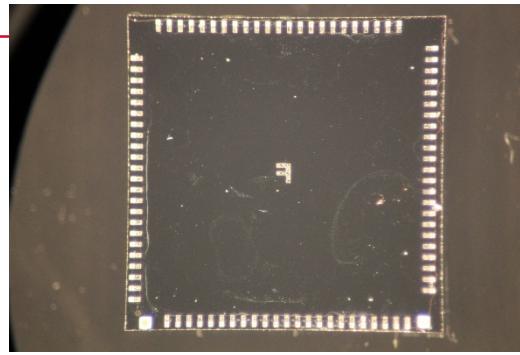
~319x
Less
silicon



~70x
Less
Silicon

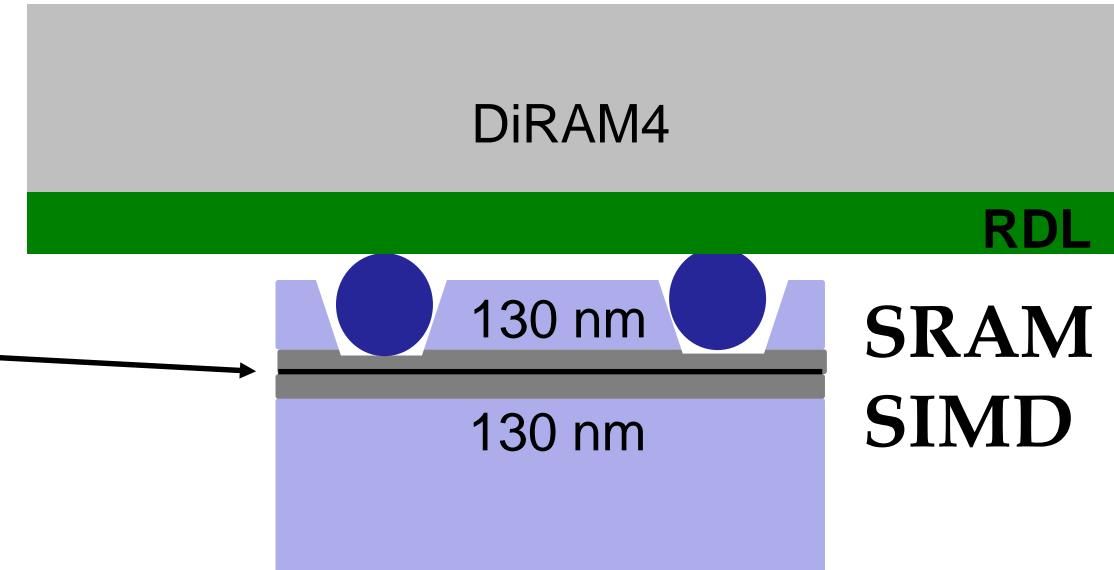
3D Implementation

- ▷ Use 3D DRAM to supply 4 Tbps memory
- ▷ Now in test
- ▷ Most features except PIM in this chip



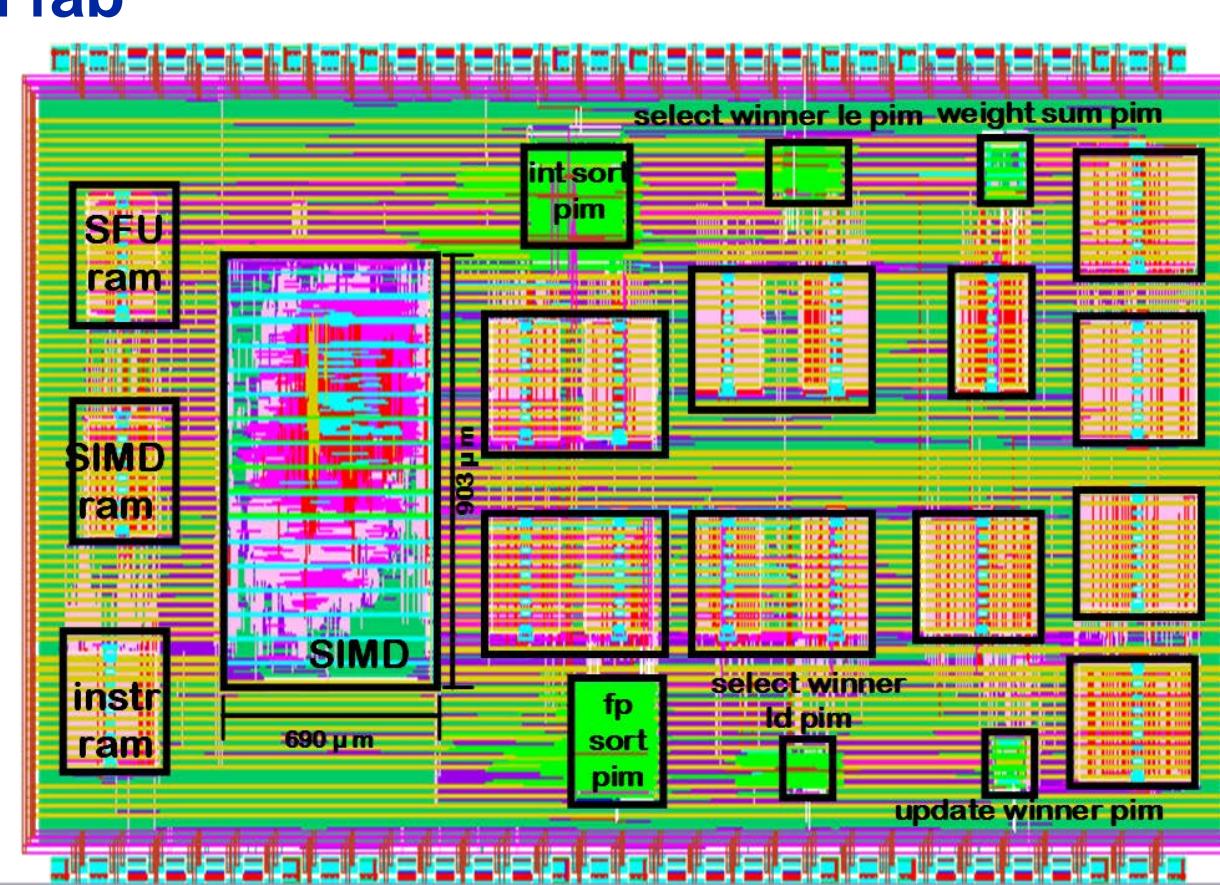
8 um

DBI

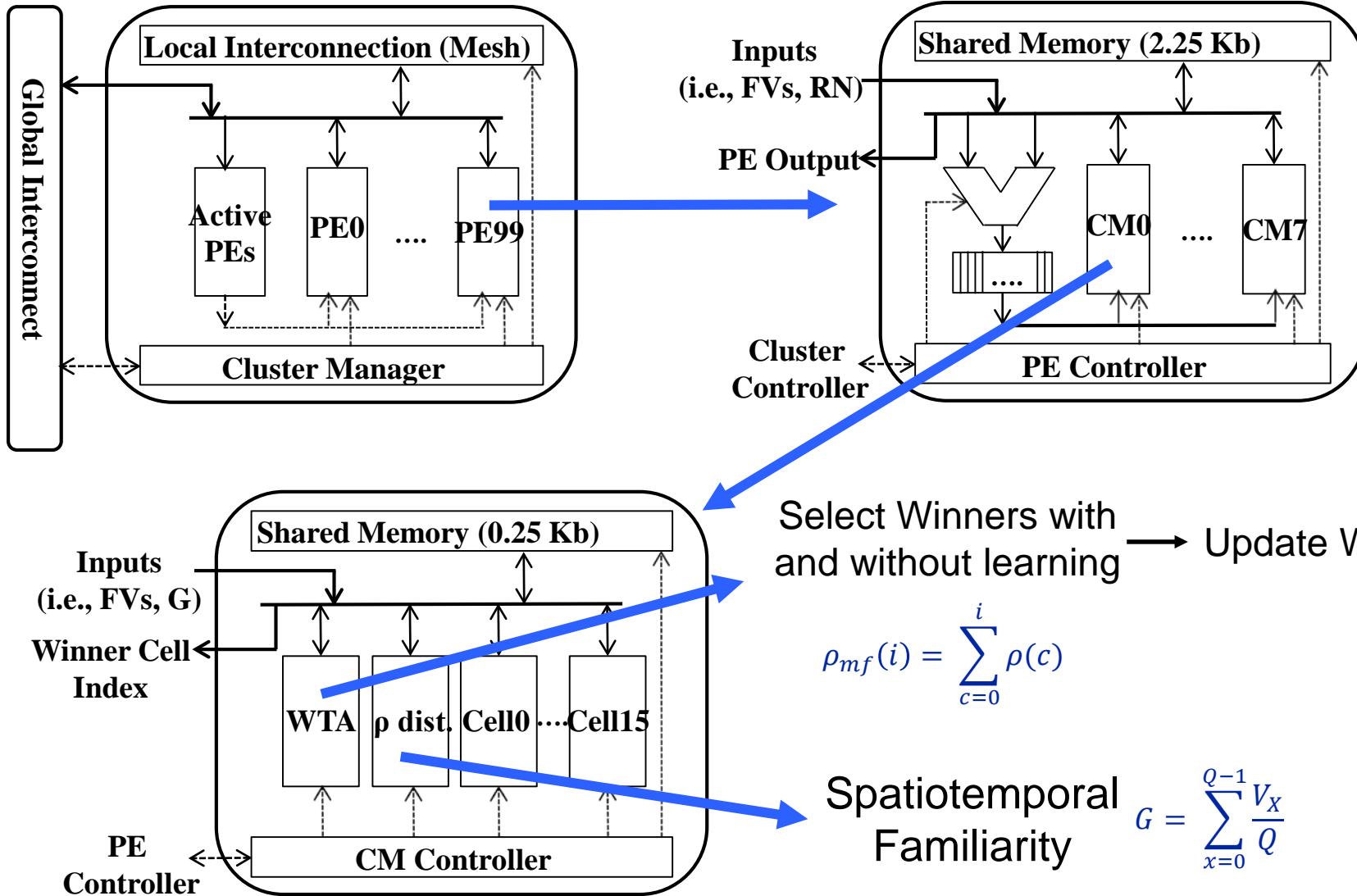


32 nm Tapeout

- ▷ Complete accelerator with ISEs and PiMs
- ▷ Still in fab



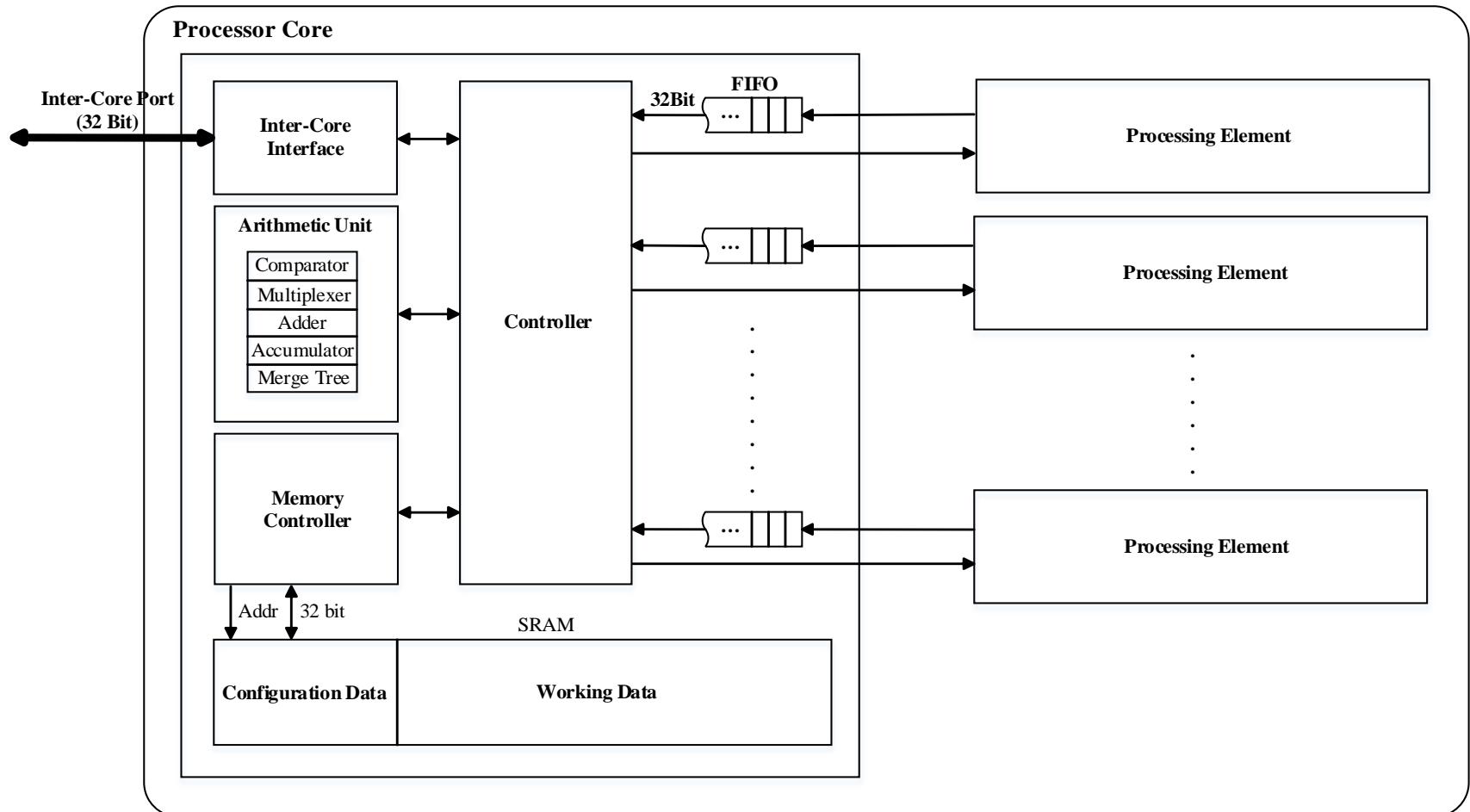
Configurable Sparsey ASIC



Sparsey ASIC

Factor	GPU Baseline	ASIC
Sparsey		
KTH Run time	8 ms (1)	0.06 ms
Power	200 W	22 W
Logic Area		220 mm ²
Performance/Power	1	1300

HTM Processor Core

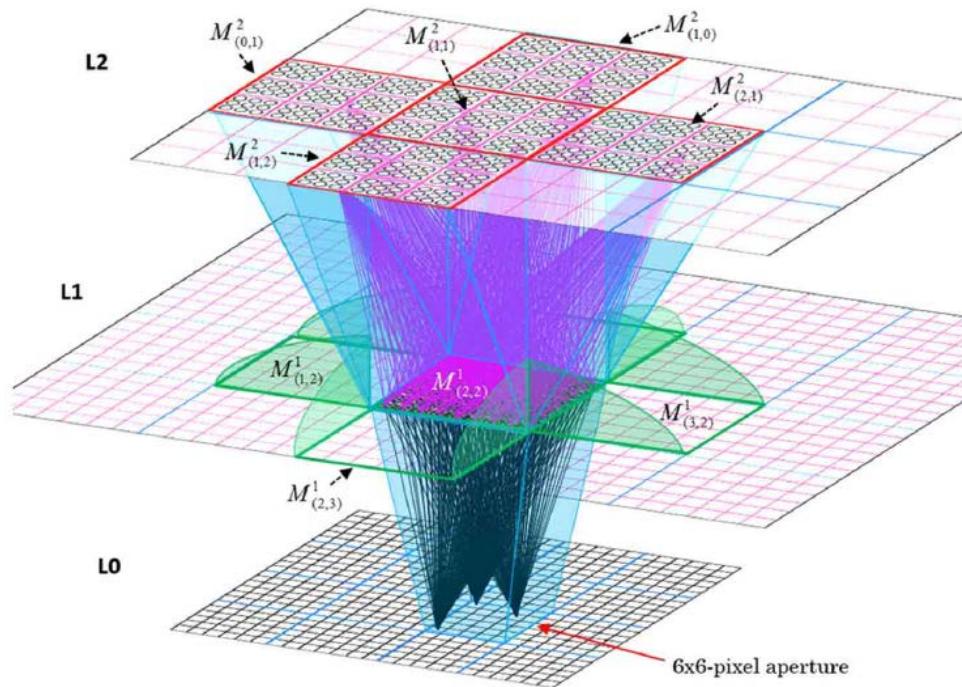


HTM ASIC

Numenta HTM	GPU	ASIC
KTH run time	225 ms	0.63 ms
Power	70 W	298mW
Logic Area		5.5mm ²
Performance/Power	1	47,100

Memory Usage

- ▶ HTM and Sparsey can work with unlabeled data and in-line
 - ◆ They essentially do this by remembering everything and organize the associativity in time and space based on the learned data



Memory Usage

- ▶ Fully customized 65 nm ASICs for Sparsey and HTM

Algorithm	Bandwidth	Capacity	Frame Rate
Sparsey	530 Tb/s	800 Mb	16000 fps
Numenta HTM	0.2Tb/ps	524 Mb	1600 fps

- ▶ SIMD with PiM Solution @ 30 fps

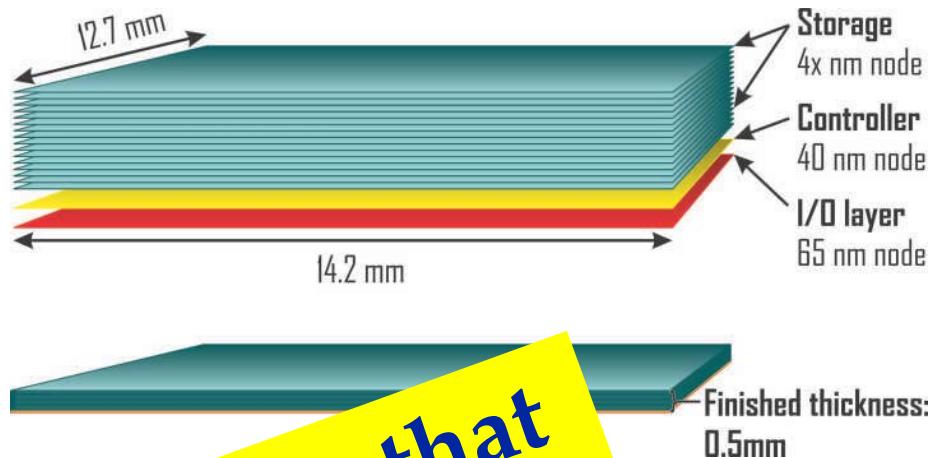
Algorithm	Bandwidth	Capacity
Sparsey	4.8 Tbps	314 Mb
Numenta	0.9 Tbps	1130 Mb

- ▶ Small by real world standards

Tezzaron DiRAM4

► “Dis-integrated” RAM

- ◆ DRAM only tiers
- ◆ Sense amps, etc. on logic only tiers
- ◆ 32 bit DQ / port



Looking at modifications that would permit 130 Tbps

Family Name	Ports	Banks per Port	Cell Size	Bandwidth	Latency
DiRAM4-64C64™	64	1	0.6 – 1.3V CMOS I/O	32 Gb	4/4 Tb/s (R/W) 9 ns
DiRAM4-64C32™					
DiRAM4-64C16™	64	64	0.6 – 1.3V CMOS I/O	16 Gb	4/4 Tb/s (R/W) 9 ns

Custom HW for LSTM

► Early Performance/Power Comparisons

Name	GPU	ASIC	X
Technology	28 nm	32 nm	
Area (mm ²)	118	54.15	2.18
Speed (ms)	93	8.5	11.0
Power (W)	65	27.3	2.38
Speed/power		26.18	

- ◆ GPU: Single precision storage and arithmetic
- ◆ ASIC: Half-precision storage and arithmetic

Conclusions

- ▷ **CPU + PiM extensions lead to (c.f. GPU)**
 - ◆ 10x – 100x improvement in performance
 - ◆ 10^3 – 10^4 improvement in performance/power
While preserving generality
- ▷ **ASIC gives another 10x in performance, and 1x to 100x in performance/power**
- ▷ **Memory capacity and bandwidth management key to success**
 - ◆ Customized Memory/PiM mating