A 3DIC system to aid in the acceleration of systems that employ multiple instances of artificial neural networks

Final oral exam: Lee B. Baker
Date: 28th February 2018

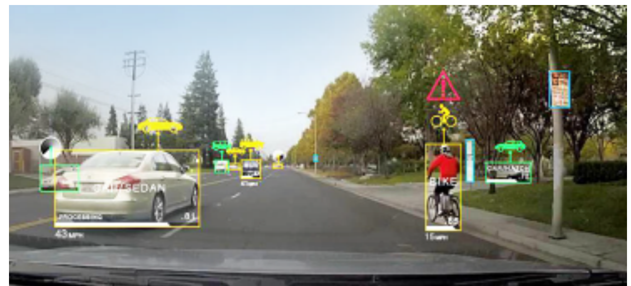
PhD Committee:
Dr. W. Alexander, Dr. G. Byrd,
Dr. P. Franzon(Chair), Dr. R. Warr

Introduction

- Recently Artificial Neural networks (NN) have demonstrated superior performance in classification and function approximation

Deep Neural Networks (DNN) have been very successful in image processing applications [Kri12]

- it is anticipated that DNNs will be employed more and more in self-driving cars

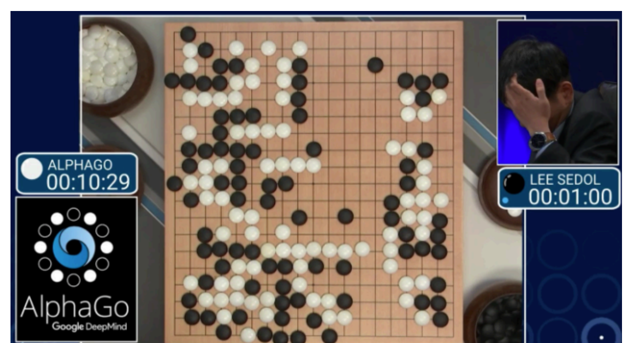


http://www.hackerboards.com

http://www.BGR.com

Reinforcement learning is gaining popularity

- alphaGo employed reinforcement learning with deep neural networks [Mad14]



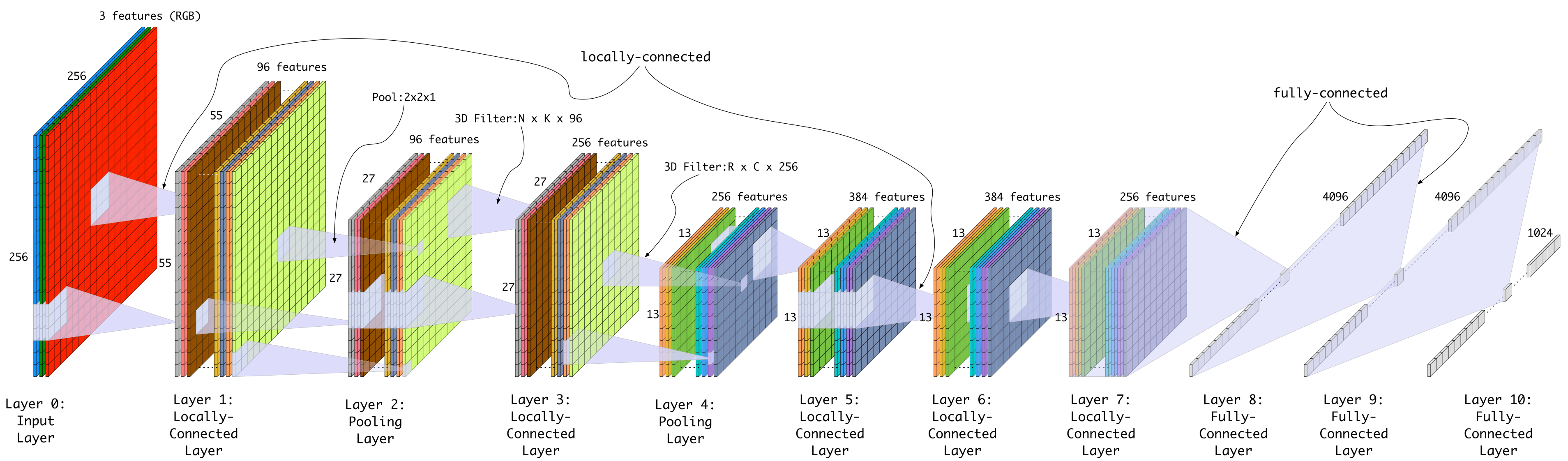
http://www.engadget.com

Introduction

- What about applications that will and do require multiple NN processors running at or near real-time?
  - drone scanning in many directions during navigation
    - looking down different than looking forward??
  - aircraft performing system diagnostics during flight
    - observing thermal profiles
    - observing vibrations
  - airport security
    - face recognition
    - body thermal profiles
    - motion

Introduction

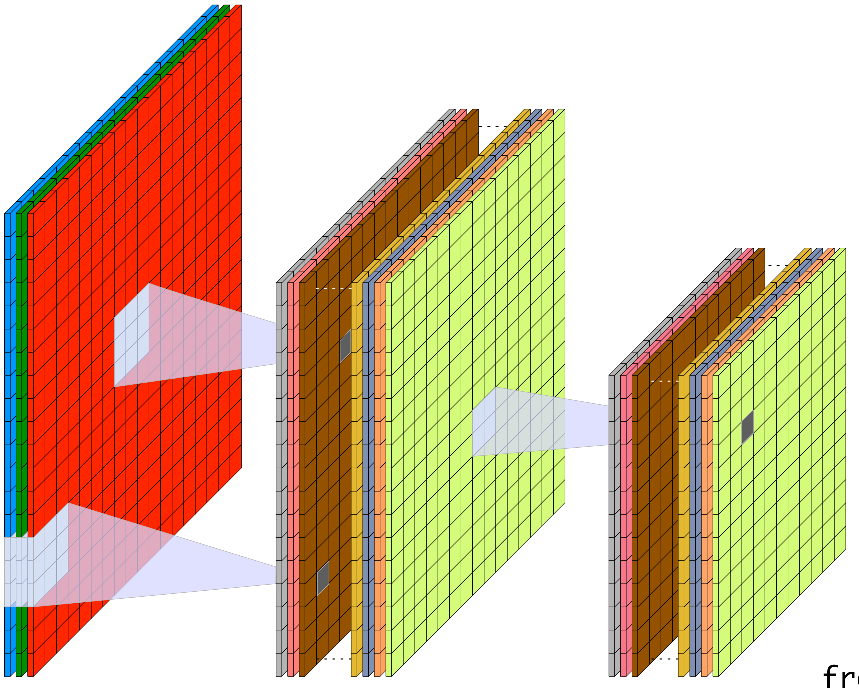
- Most useful ANNs are big
  - 100's of thousands of artificial neurons
  - real-time processing requires large amounts of memory bandwidth and storage
  - DRAM is required
  - Most implementations employ SRAM for local memory



from [27]

ANN Types

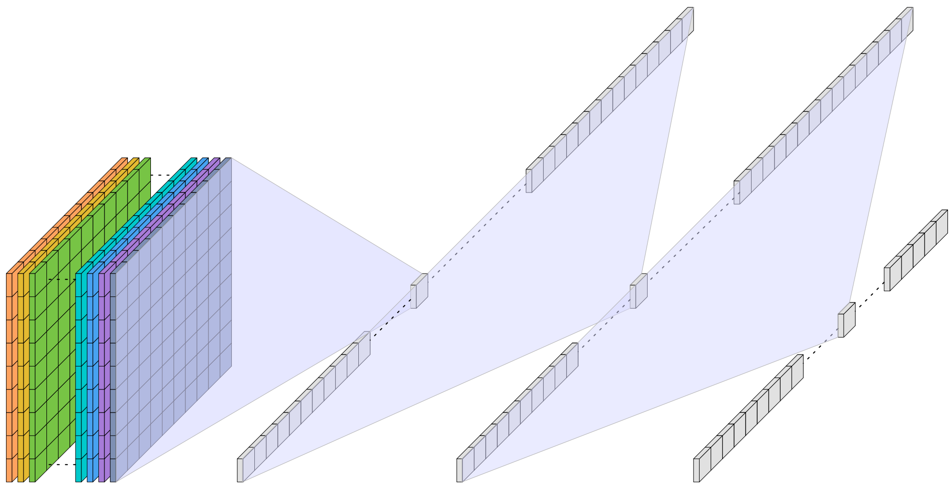
- Convolutional neural networks
  - shared filter kernel allows reuse
  - classifier stages are fully-connected – no reuse
  - non-shared kernels – no reuse [42][27]

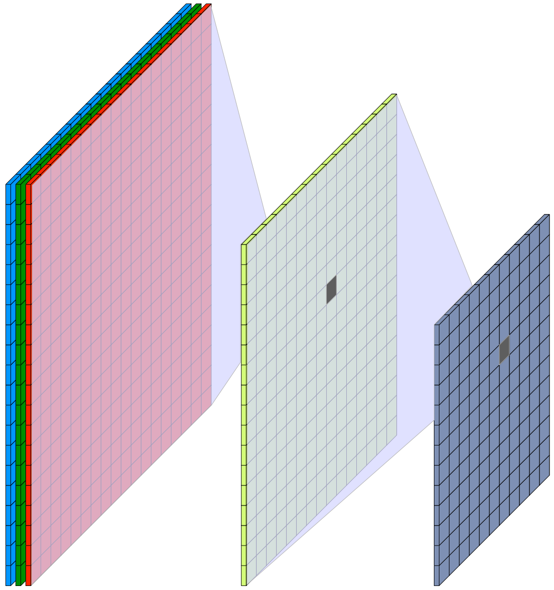


from [27]

ANN Types

- DNNs and LSTM
  - fully-connected - no reuse
  - classifier stage in CNNs





from [27]

Research

- Most research focused on CNNs
  - can take advantage of SRAM using parameter reuse
- A lot of focus on server applications
  - perhaps lots of research $$ available??
  - server applications can take advantage of SRAM for batch processing
- But many DNNs are fully-connected [1]
  - LSTM and MLPs
  - classifier stage of all ANNs including CNN
  - CNNs represented only 5% of cloud workload [1]

Mission Statement

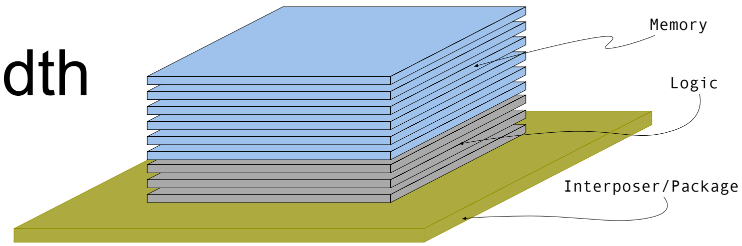
- Real world embedded applications will require multiple disparate NNs to be solved simultaneously
  - If ANNs fulfill their potential, they will be used for various system functions
- Need the capacity provided by DRAM
  - Avoid dependency on SRAM
  - SRAM is easy to use, but capacity limitations impose ANN size restrictions.
- Need to consider the system impact
  - Research often focuses on point problems. Need to consider interaction between blocks.

Problem

- Many embedded applications will/do have power, space and weight limitations
  - to achieve near or at real-time processing, current solutions require high power and high real-estate
    - GPU solutions large and high power
    - ASIC's better than GPUs
  - Both have memory bandwidth or capacity limitations when reuse opportunities do not exist

Solution

- 3DIC Architecture
  - reduces energy and area
  - increase connectivity and bandwidth



- 3D-DRAM
  - provide high bandwidth and large storage
  - operate directly out of DRAM
- Data Structures
  - data structures to ensure neural network data can be accessed efficiency
- Specialized processing layers
  - provide special functions to aid in acceleration of target neural networks

Feasibility

- Can a 3D-DRAM be used effectively?
- Can a useful system fit within the 3D stack footprint?
- How can we control such a system?

Contribution

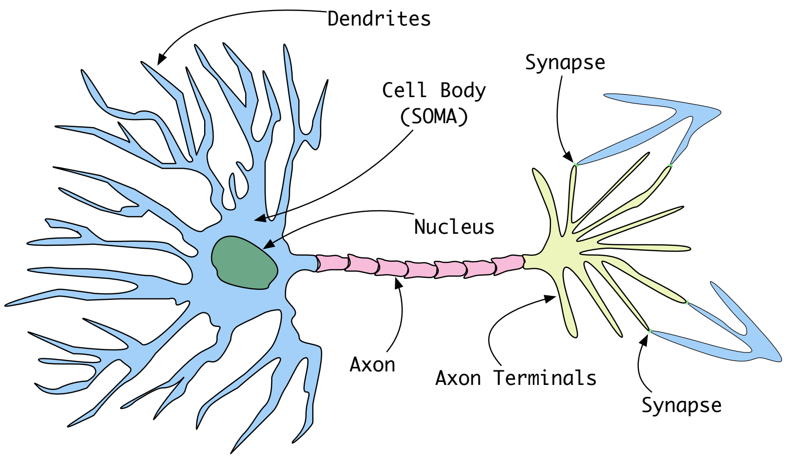
- An extensible architecture that can simultaneously process multiple disparate real-time DNNs
- Proposed a custom 3D-DRAM providing a ~32X bandwidth benefit compared to standard 3D-DRAM
- A DNN system solution that employs pure 3DIC technology
- Custom instructions and data structures that facilitate operating directly out of 3D-DRAM

Outline

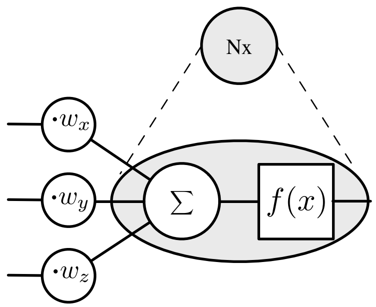
1. What are artificial Neural networks
2. Target application and ANN type's
3. State-of-the-art
4. System Architecture
   - Problem
   - Solution
5. System details
   - 3DIC DRAM and data structures
   - 3D Bus
   - Manager
   - Processing engine
6. Results
7. Summary

Artificial Neural Networks

- a network of processing elements inspired by the connectivity and processing observed in the brain

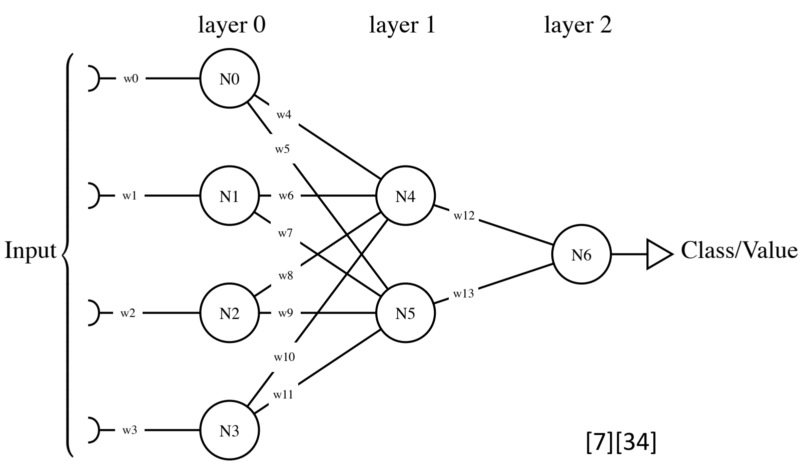


[7][34]



[7][34]

- the processing elements or neurons are connected together to form a network



[7][34]

Artificial Neural Networks

- the processing elements fall into two categories
  - rate based[7][34] neurons which try to capture the neuron behavior in the form of a number
    - relatively simple to model
    - easier to train and have shown high levels of efficacy in many applications
  - spiking neurons[7][34] which more closely emulate actual brain behavior
    - model neural activity in the form of differential equations
    - require numerical methods to solve although there have been attempts to employ analog circuits
- In this work we are focusing on rate based ANNs

Summary : Target application and ANN type's

- Embedded systems
  - not cloud based
  - assumes multiple tasks are being performed
  - requires multiple disparate ANNs
  - inference only
- This work has focused on support for:
  - fully- and locally-connected DNNs
  - can be extended to support LSTMs

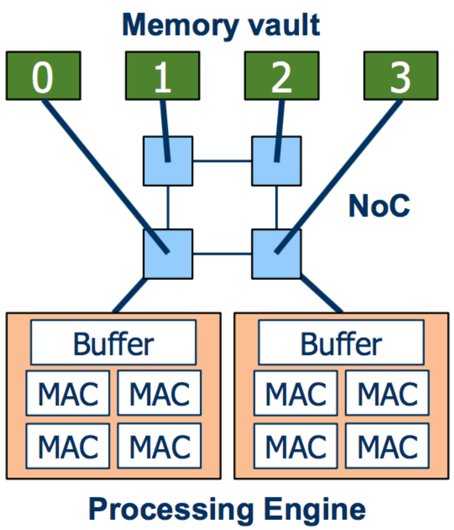
State-of-the-art

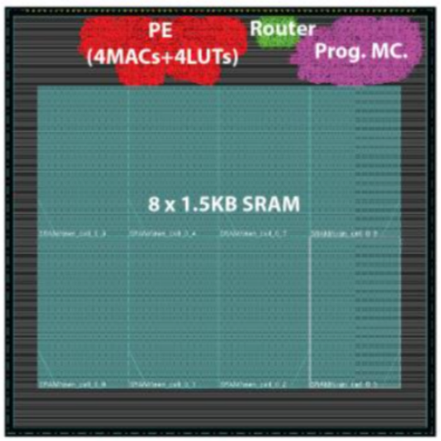
- Current implementations either use GPU's or low capacity ASIC implementations
- GPU's are typically general purpose devices and consume high power
  - GPU's do lend themselves to convolution when weights are shared
  - Lemma: ASICs always perform better than GPUs
- ASIC implementations target larger networks by using multiple devices
  - usually local SRAM which limits network size
  - some/most implementations target specific NN's, usually CNNs

State of the Art ASIC's

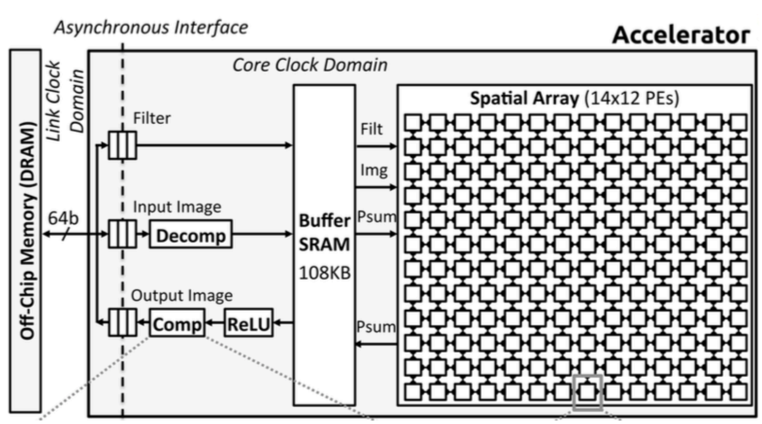
- ASIC's

NeuroCube[24]





Eyeriss[11]



State of the Art
ASIC's

- NeuroStream[3]
  - acknowledges DRAM is required for useful DNNs
  - uses HBM 3D-DRAM along with local SRAM
  - depends on SRAM for performance
- Google TPU[1]
  - assumes large levels of batch processing
  - 8-bit processing
  - fully-connected NNs represent very high percentage
  - acknowledges performance degradation using fully-connected
- DaDiannao[10]
  - uses embedded DRAM but still requires up to 64 for useful DNNs
  - high internal bandwidth but dissipates high power when scaling to useful-sized ANNs

[1] [Kim16], [2] [Che16], [3] [Esm05], [4] [Far11]

State of the Art

- Current state-of-the art ASIC's and GPU's do not provide the capacity and bandwidth to provide a solution requiring multiple useful disparate ANN's to be computed in real-time
- Goldilocks principle
  - need balance of bandwidth and storage
  - need to read all parameters in sample time for real-time processing
  - Currently SOA bandwidth too low or too high

System Requirements

- System solution requiring multiple NN's
  - binary32 number format
  - ~8GB of memory for baseline ANN
  - real-time processing : ~16mS (60 frames/sec)
  - ~26Tbps memory bandwidth for baseline ANN
- Current 3D-DRAM memory technology bandwidth
  - HMC – SERDES~2Tbps – too low
  - HBM – wide DDR ~2Tbps – too low
  - Tezzaron DiRAM4[14] – wide DDR – 4Tbps – too low
- Propose Customized DiRAM4
  - expose more of the page
  - take advantage of high density TSVs

Architecture Blocks

- Proposed Customizations to standard DiRAM4 3D-DRAM
- Data Structures and Instructions to support efficient use of DRAM
- Management Layer for configuration and control
  - DRAM controller
  - Instruction decoder
- Processing layer targeted toward DNNs
  - streaming functions operate directly on data
    - multiply/accumulates ANe activation
    - multiply for softmax
  - additional special functions in SIMD used in the neuron activation process
    - ReLu for ANe activation
    - add, divide and exponent for softmax function
    - compare for pooling

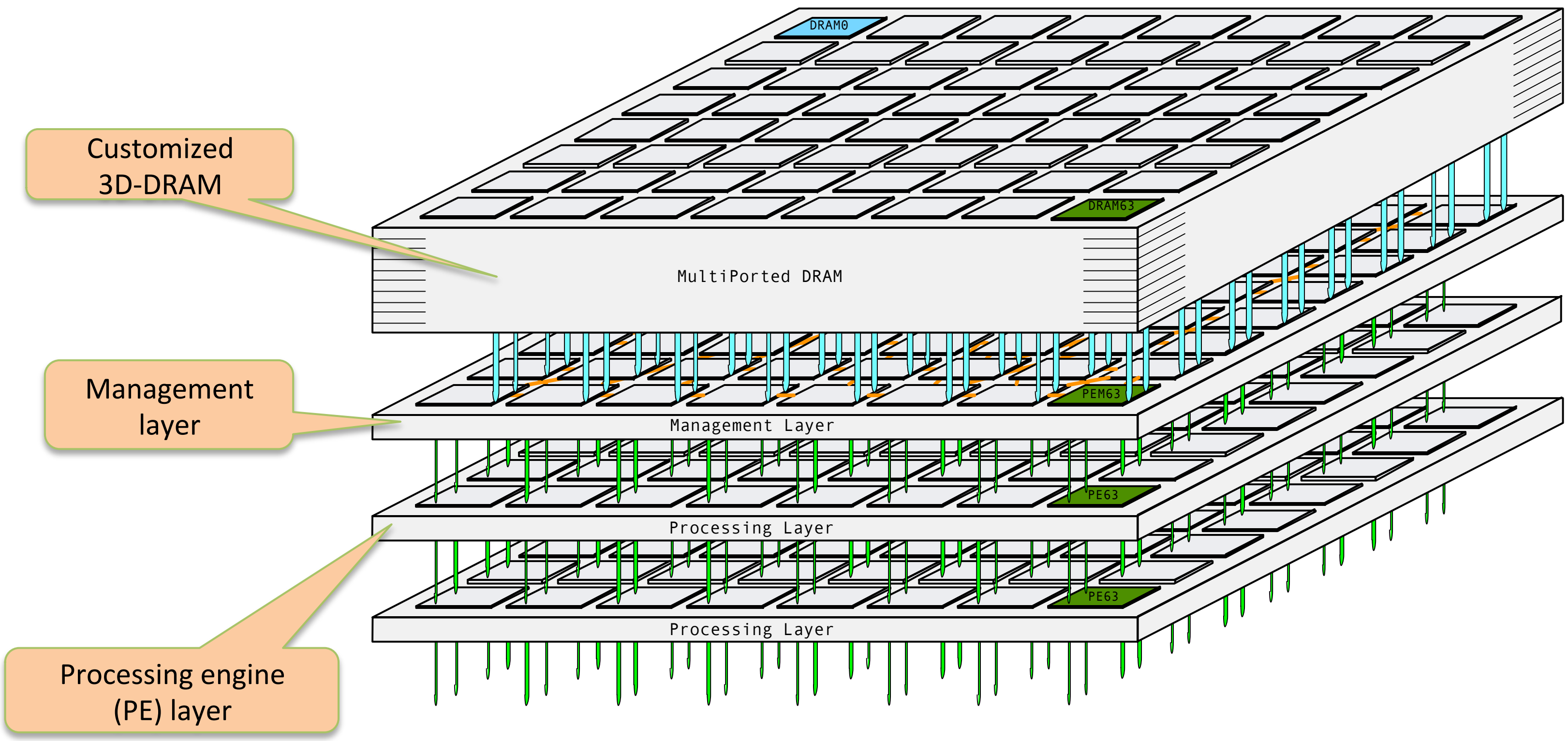
Main memory

- The 3D-DRAM is the Tezzaron DiRAM4
  - 64 disjoint 1Gb memory ports
  - System has 64 sub-systems each operating on one memory port
- Suggested Customizations for a 3D-DRAM
  - Standard DiRAM4 port is 32bits @ 1GHz
  - We suggest widen to 2048 bits and use high-density TSVs
  - Entire page in one access using burst-of-2
  - Raw bandwidth ~2Tbps per port

System Solution

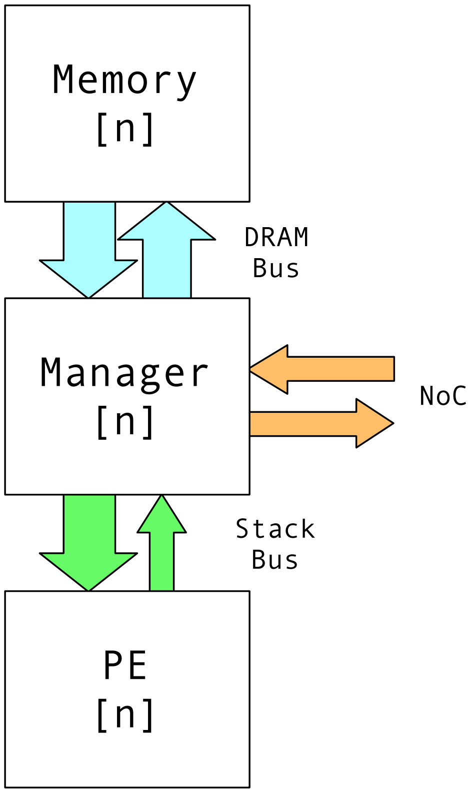
- System has 64 sub-systems (SSC) each operating on one memory port
  - ANNs split over sub-systems
  - SSCs communicate over local NoC in each Manager
- Sub-system (SSC) has a Manager and PE
  - Instruction Decode in Manager
    - contains information on how to process a group of ANes
  - Wide Data-path between DRAM, Manager and PE
    - data for 32 execution lanes with two arguments per lane
  - PE performs ANe state operations
    - process a group of ANes
    - able to perform a MAC operation between a weight and ANe state every cycle

3D Physical Configuration

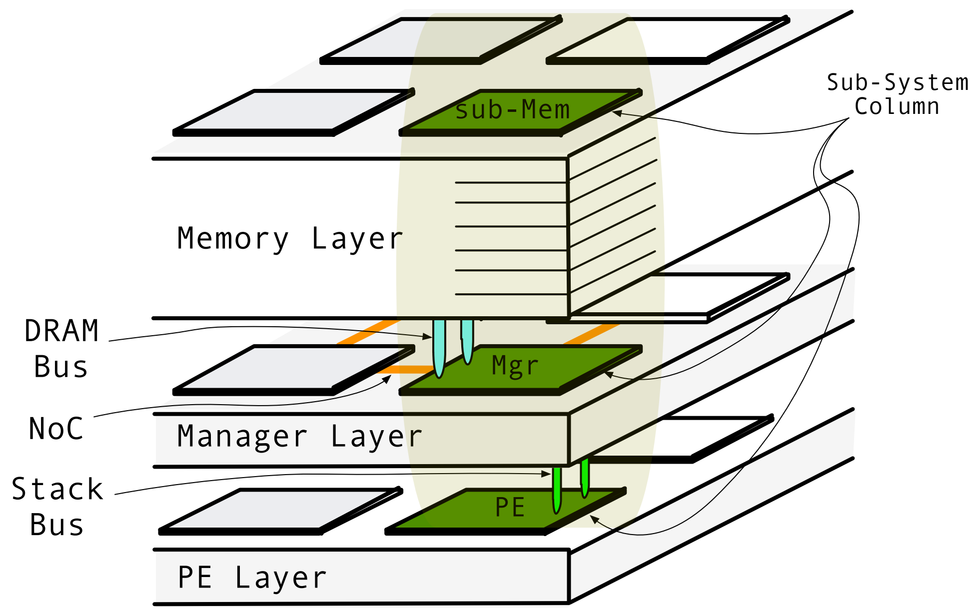


Sub-System Column (SSC)

- Consists of DRAM, manager and processing engine (PE)

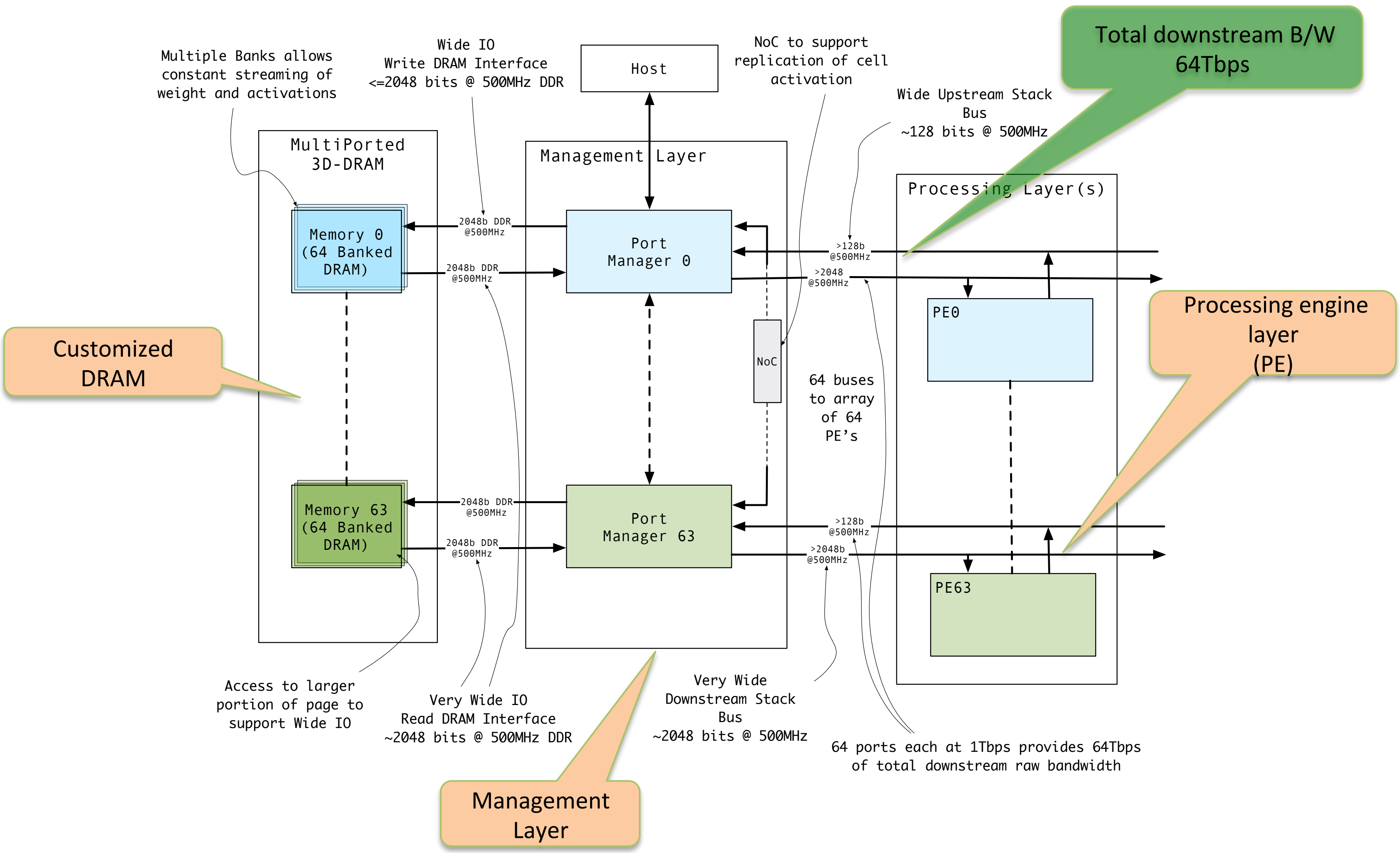


logical

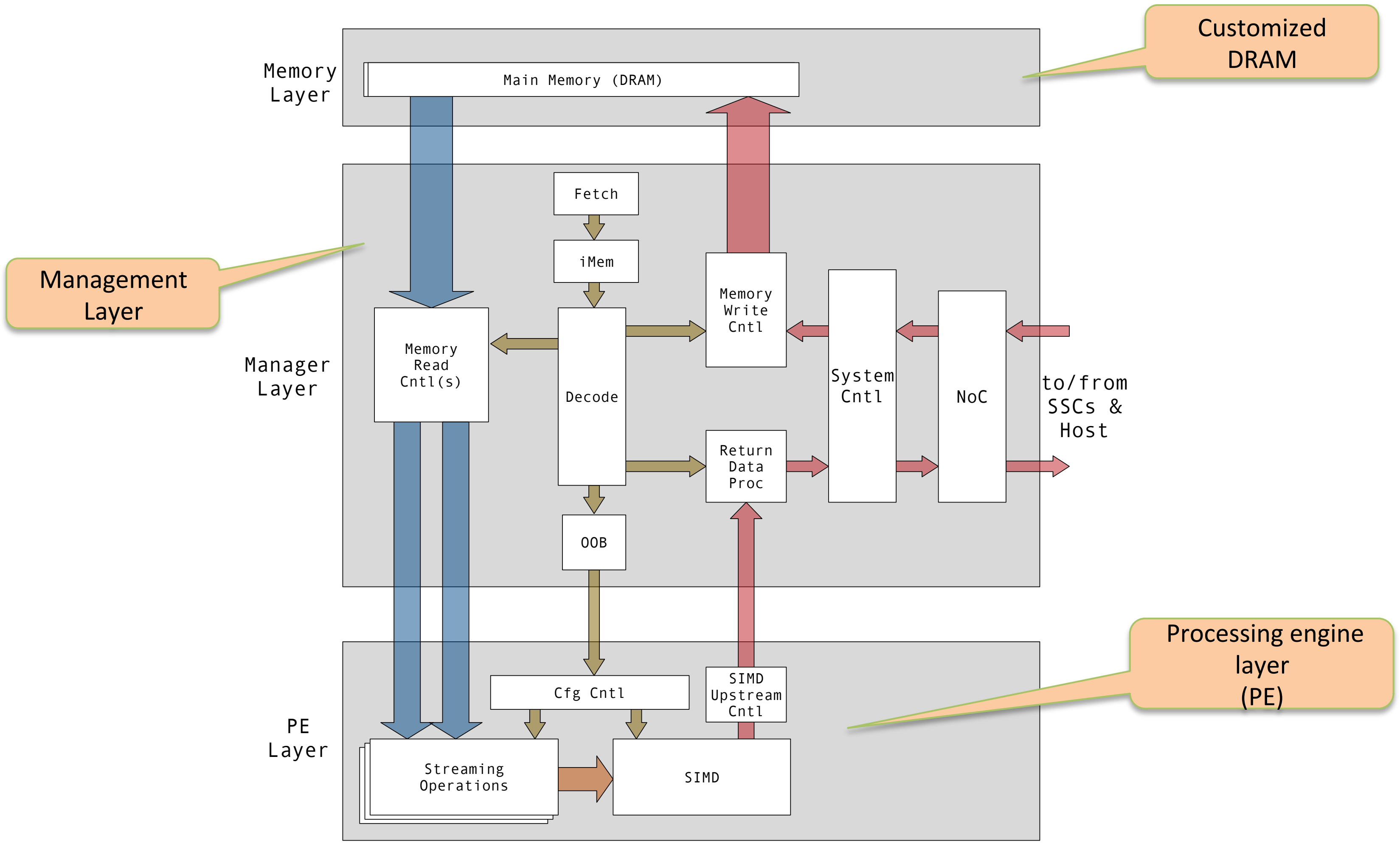


physical

Solution Block Diagram



System Connectivity



Architecture Requirements

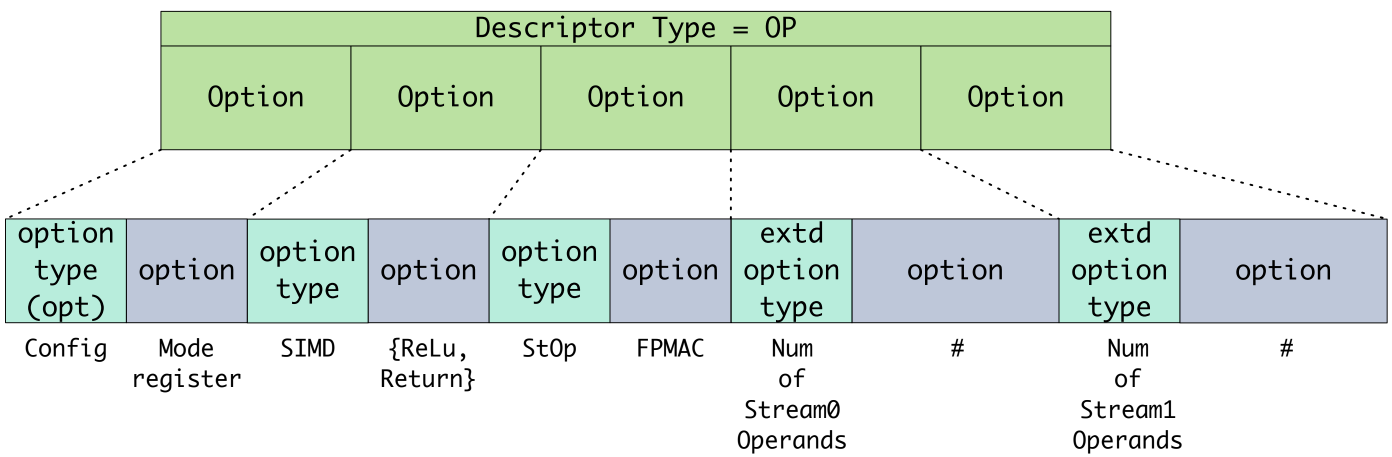
- Need to absorb DRAM latency
  - separate memory requests from read/write data
- Instruction spawns multiple commands to dependent tasks
  - concurrently pre-fetch data, prepare PE, prepare Return Data Processor
  - all operations have an associated tag
- Need to describe parameter and state storage
  - assume input stored in row-major fashion
- Network-on-Chip
  - needed for ANe state replication to all dependent SSCs

Instructions

- Instructions consist of descriptors

Operation Descriptor	arg0 Read Descriptor	arg1 Read Descriptor	Result Write Descriptor

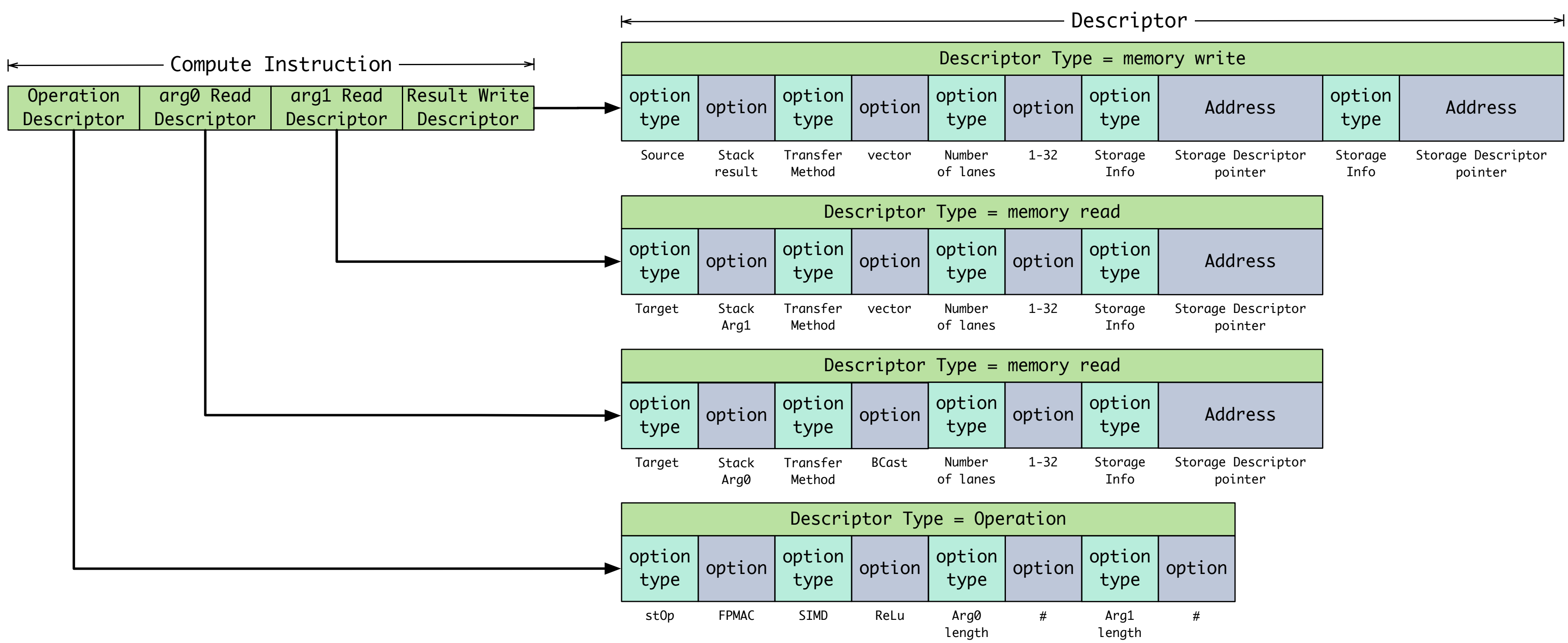
- Descriptors used to control modules which take part in the instruction
- Descriptors contain tuples which contain the details



Instruction Decode

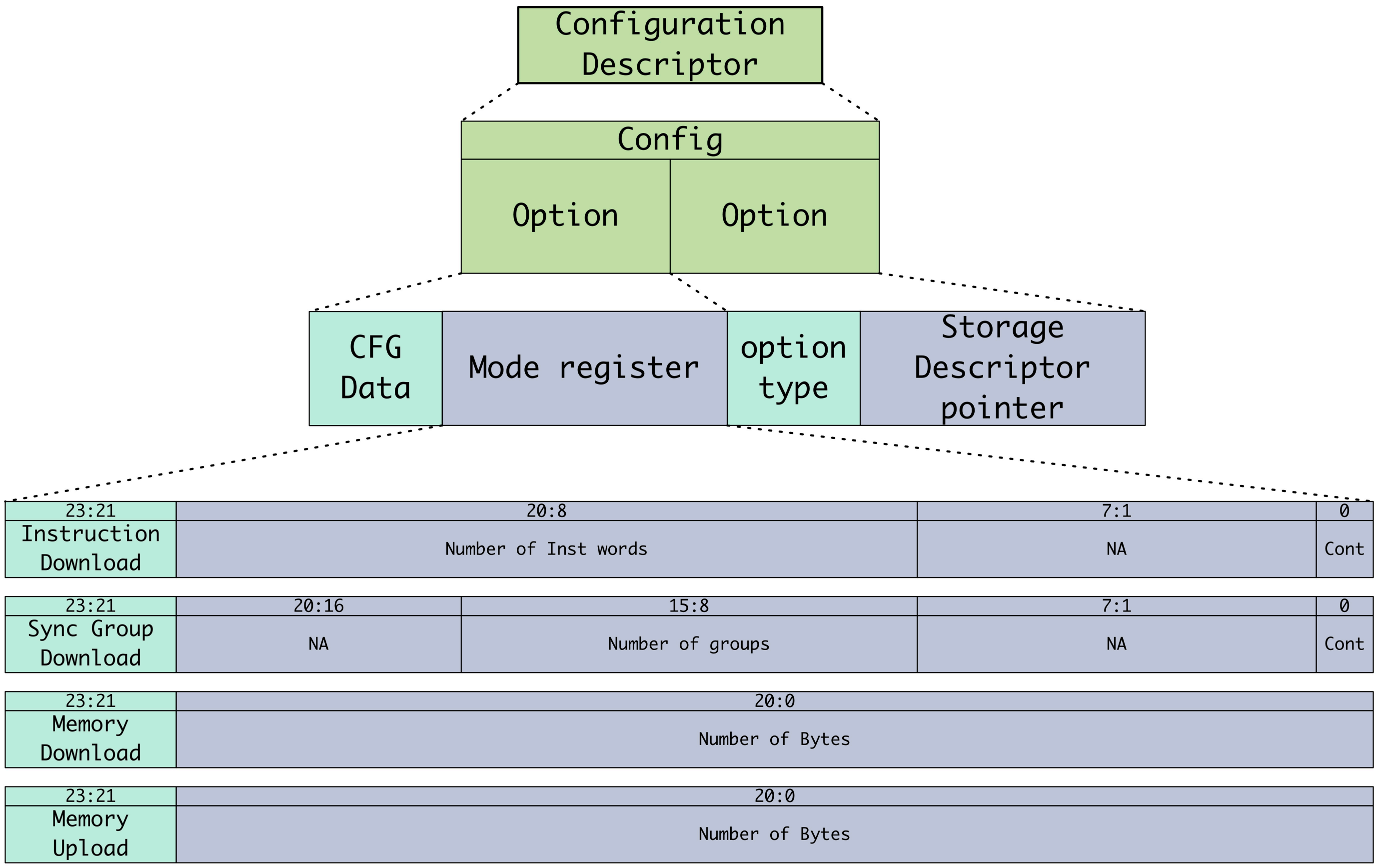
- Two types of instruction
  - Configuration
  - Compute
- Manager sends descriptors to dependent block(s)
  - may not parse descriptor if not relevant to manager
  - manager knows descriptor dependencies and sends to modules participating in instruction
- Configuration Instruction
  - send descriptor to manager control block
  - used for synchronization and data transfers
- Compute Operation
  - send descriptor(s) to blocks participating in ANe state computation
  - OP
    - send descriptor to PE
  - Memory Read
    - identify target execution lane and send descriptor to one of two memory read controller
  - Memory Write
    - send descriptor to return data processor

Compute Instruction



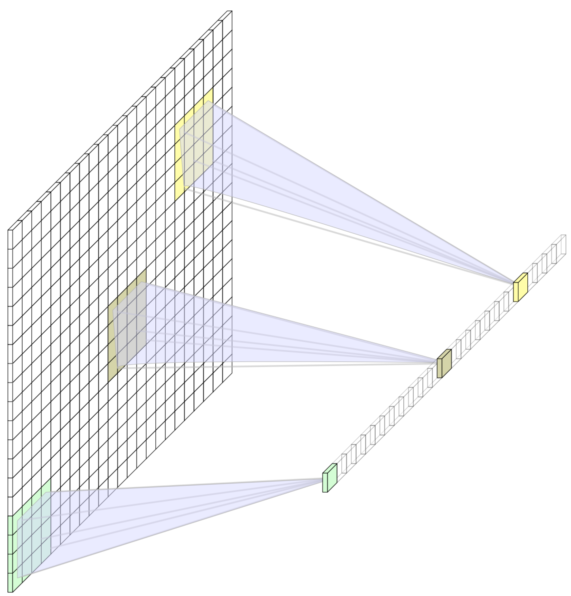
Configuration

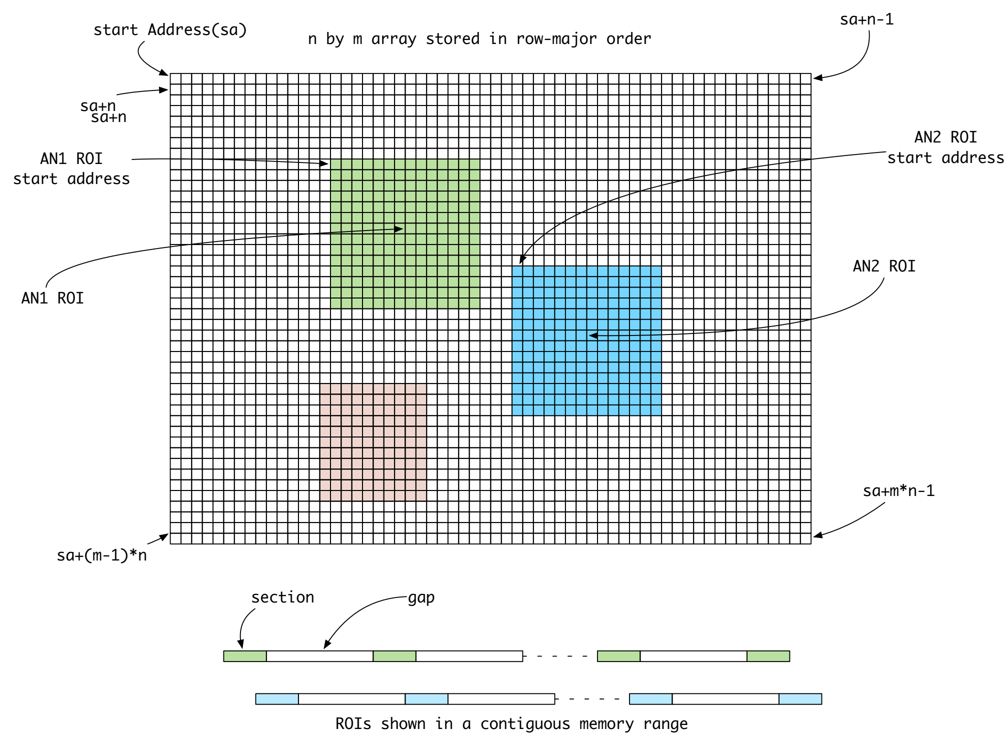
- Supports both synchronization and data transfer



Data Storage and access

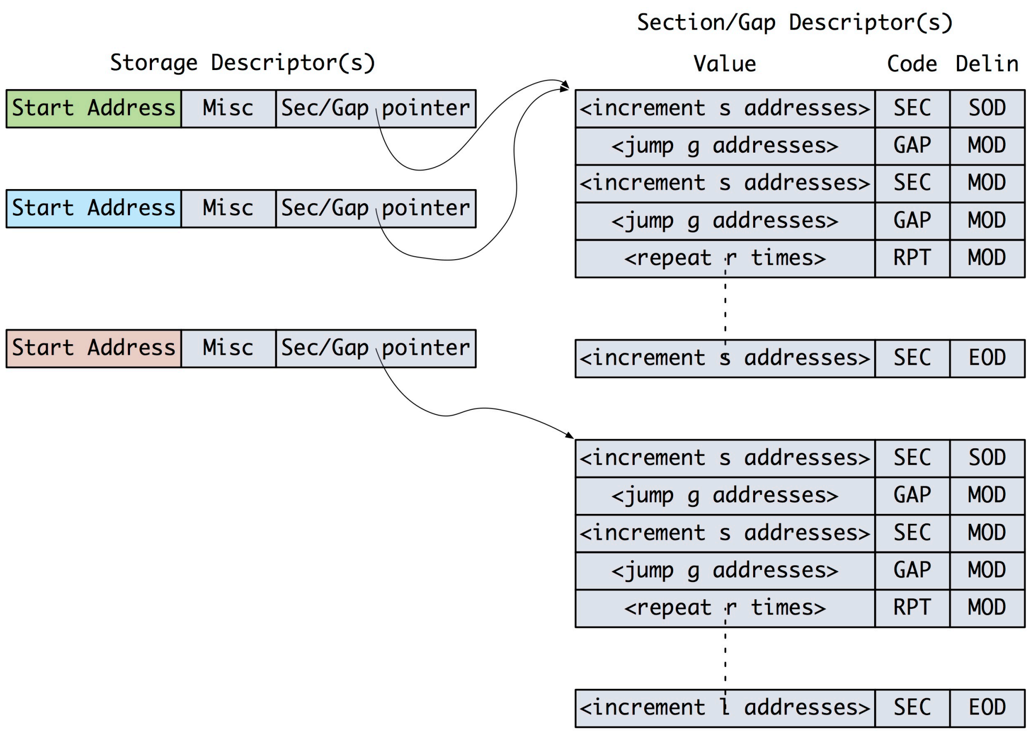
- Inputs and ANe states stored in row-major
  - two address increment mechanisms
- Weights for a group of ANes are interleaved





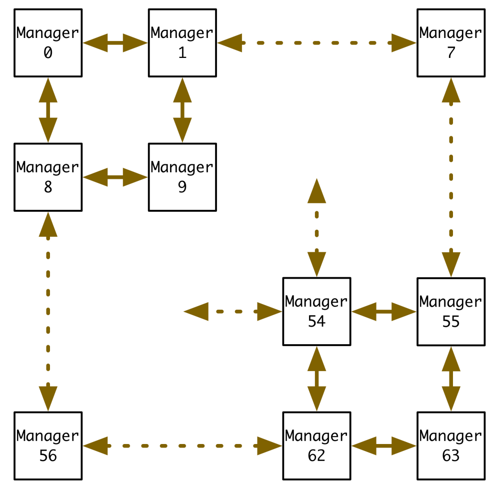
Storage descriptors

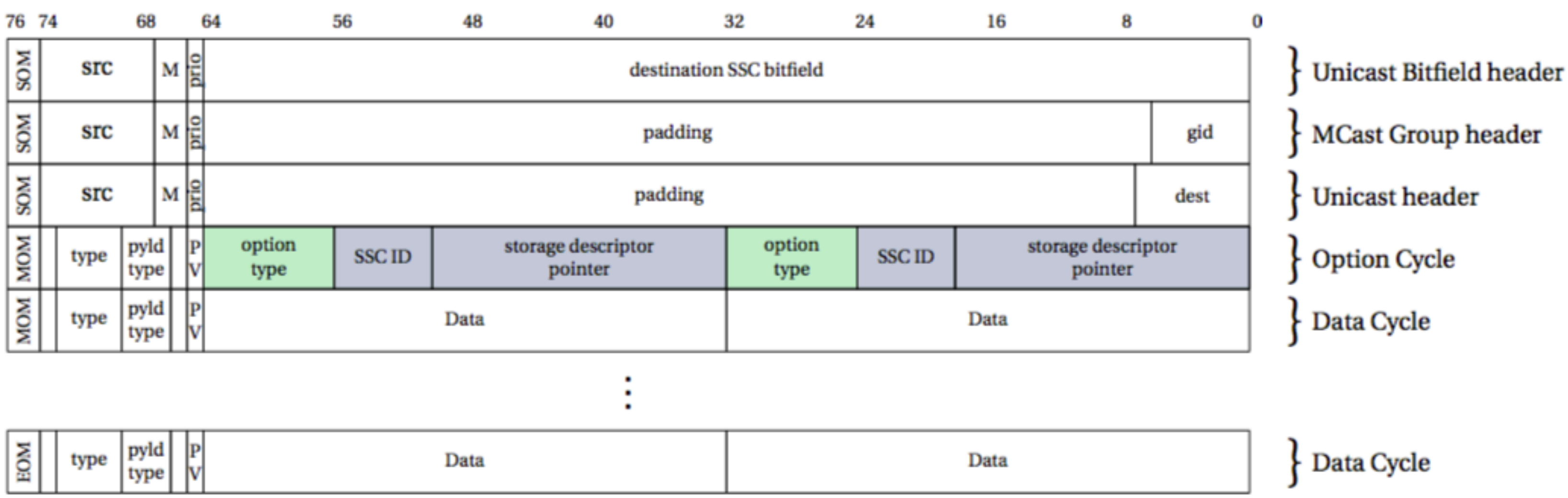
- Used to describe how data is read/written
- Used for both parameter and ANe state storage
  - instruction contains pointer to storage descriptor
  - each SSC has a set of storage descriptors
  - instruction/NoC carry pointer, not address, length etc.



NoC

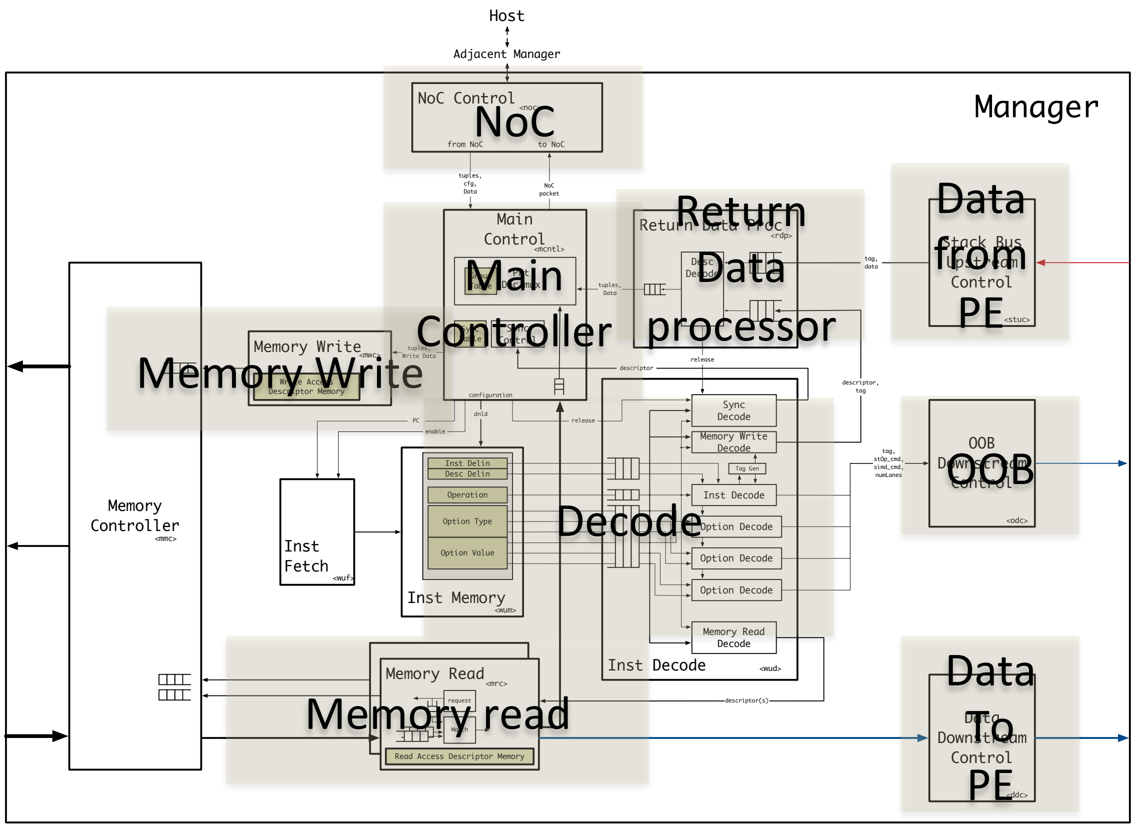
- Four ports
  - connections to adjacent SSC
  - fixed routing table
  - can be reconfigured based on performance
  - good area for additional research





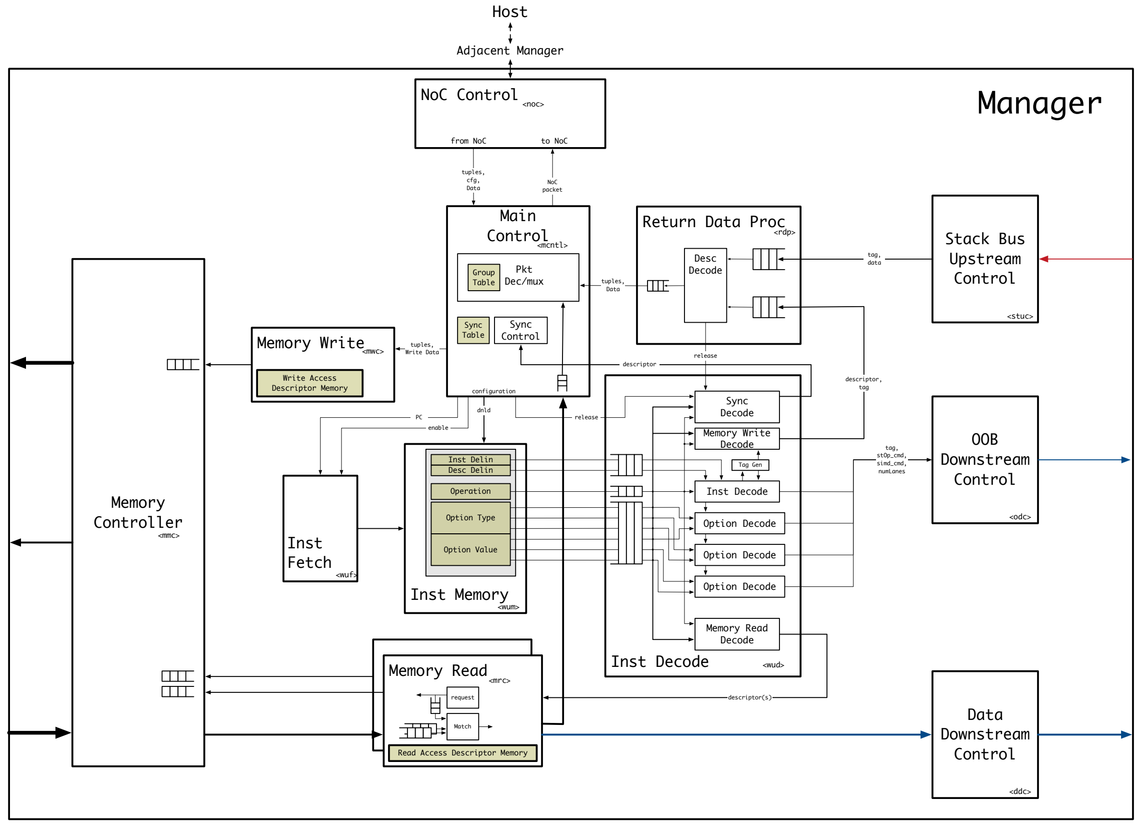
Manager Layer

- Decodes instructions
  - sub-descriptors are sent to dependent blocks
- Reads and writes to main memory
- Communicates to host and other SSCs



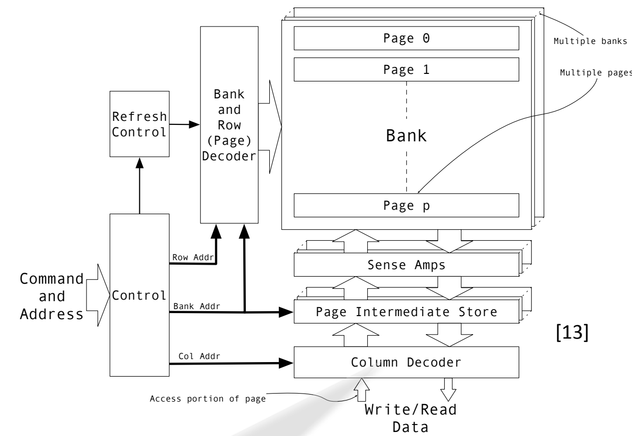
Manager Layer

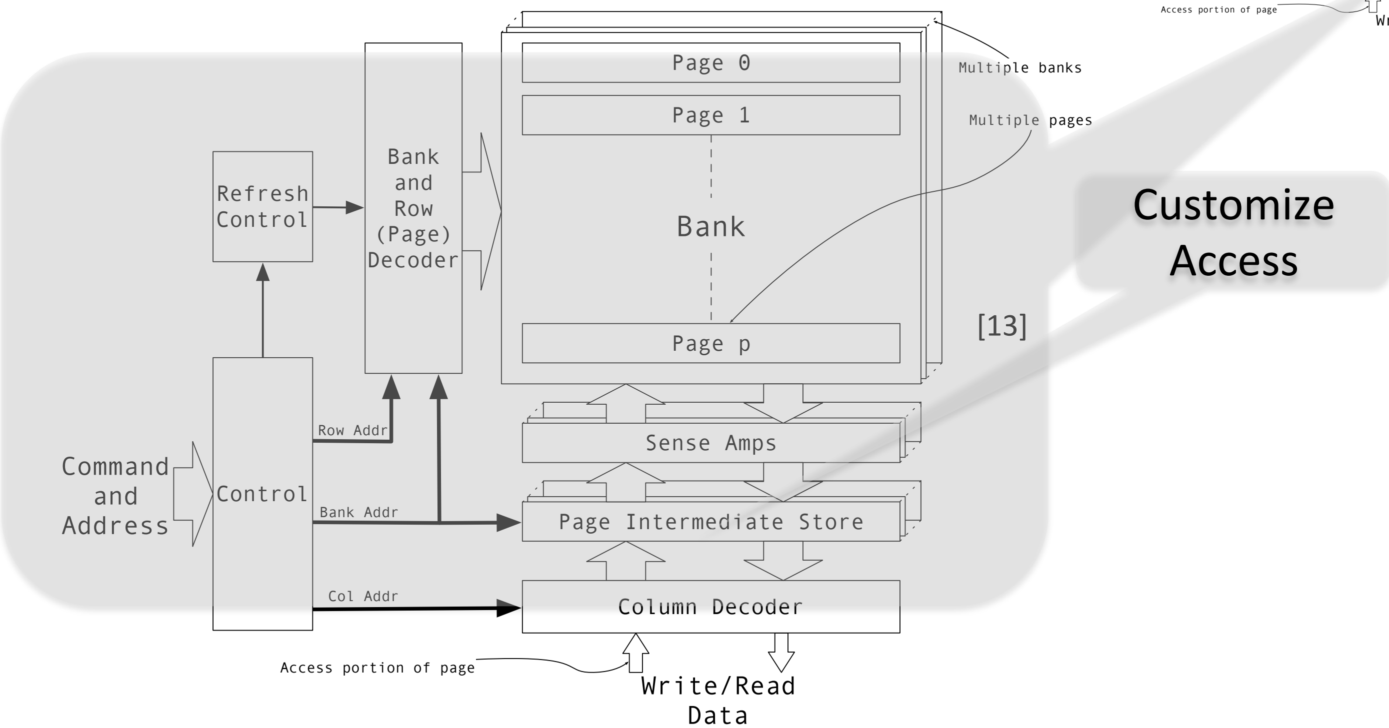
- Decodes instructions
  - sub-descriptors are sent to dependent blocks
- Reads and writes to main memory
- Communicates to host and other SSCs



DRAM Customizations

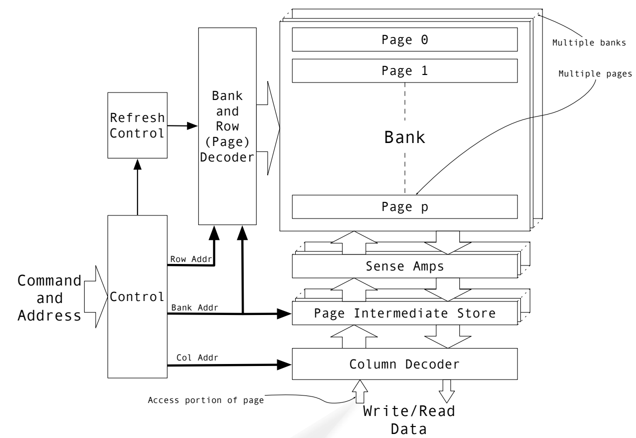
- Expose more of the page
  - Each burst-of-2 read transfers entire page
  - Need to accommodate idle time during Page Close
  - Employ TSVs for wide bus
- Provide write masks
  - To avoid Read-Modify-Write

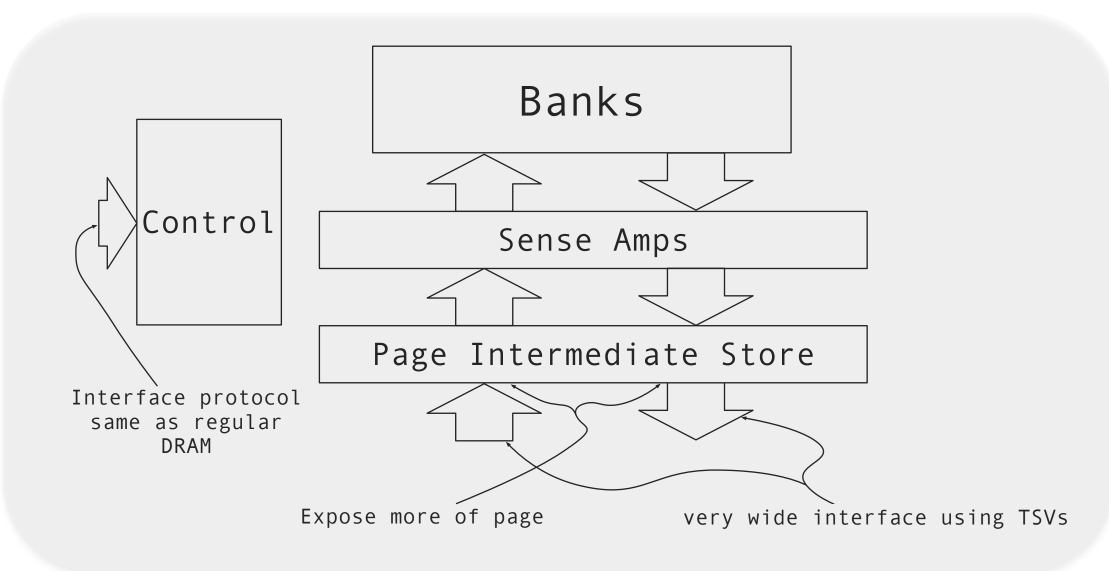




DRAM Customizations

- Expose more of the page
  - Each burst-of-2 read transfers entire page
  - Need to accommodate idle time during Page Close
  - Employ TSVs for wide bus
- Provide write masks
  - To avoid Read-Modify-Write

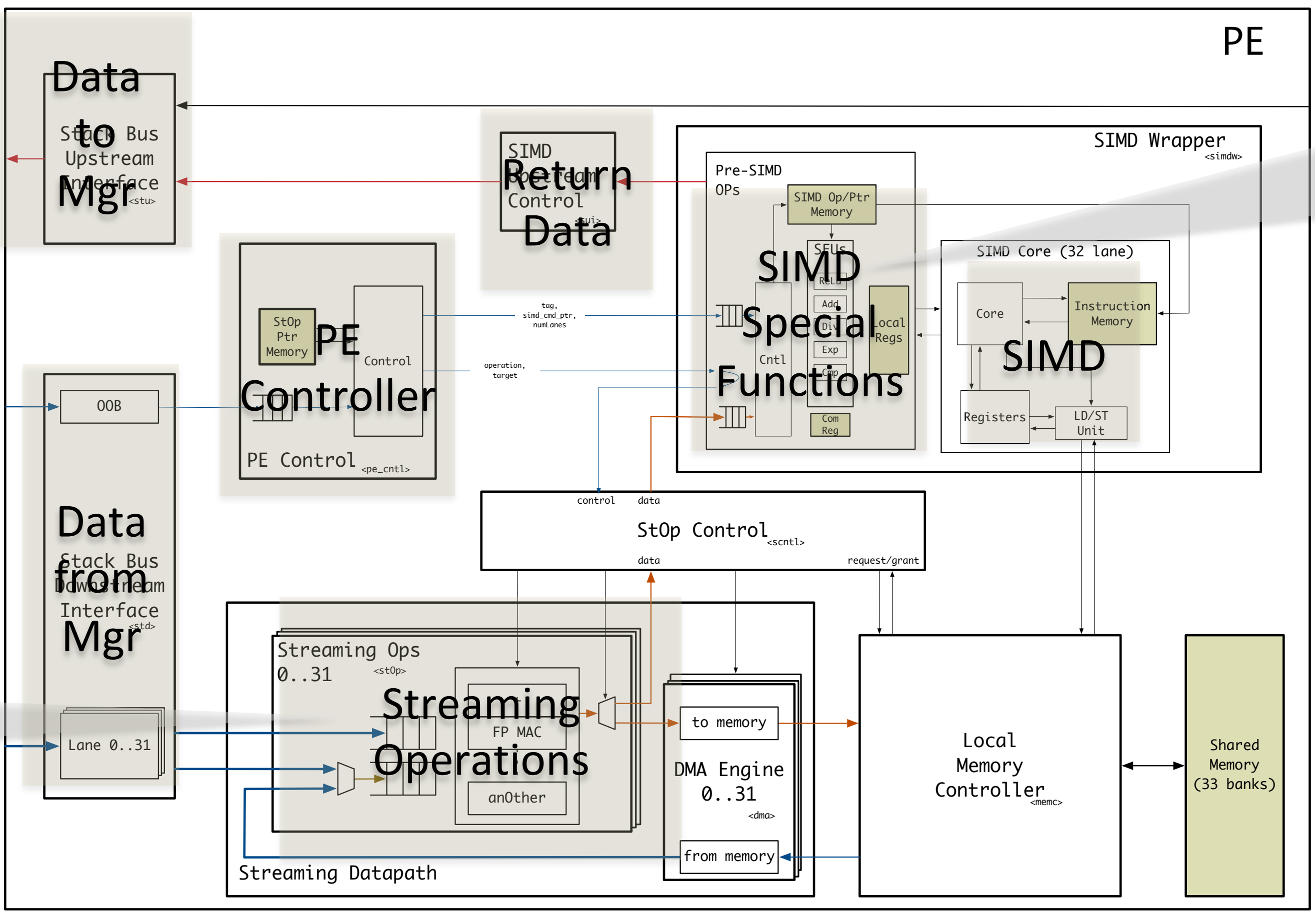




Processing (PE) Layer

- Streaming operations operate directly on data
- SIMD Wrapper performs post stOp tasks

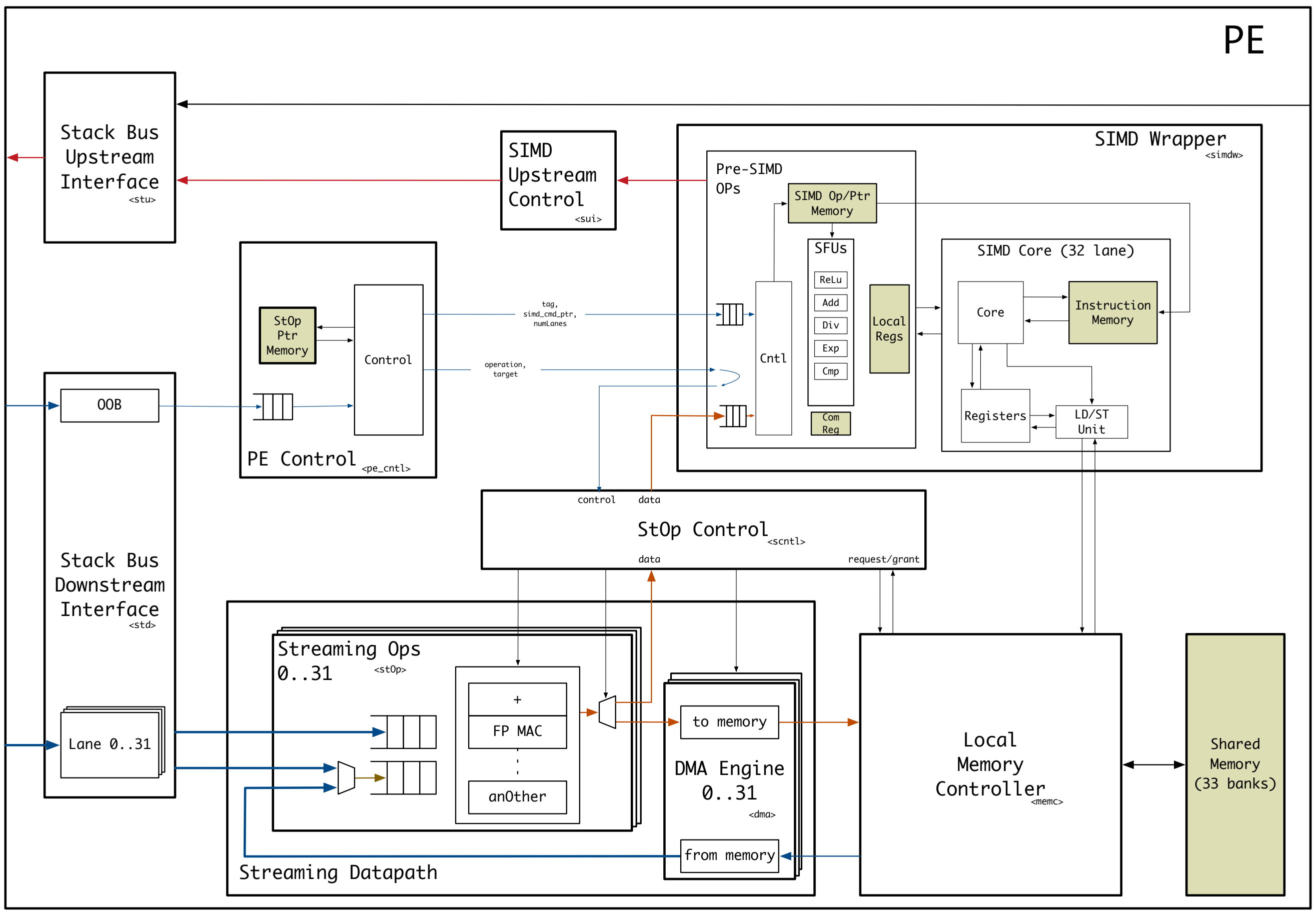
ReLu
Add
Divide
Exponent
Compare



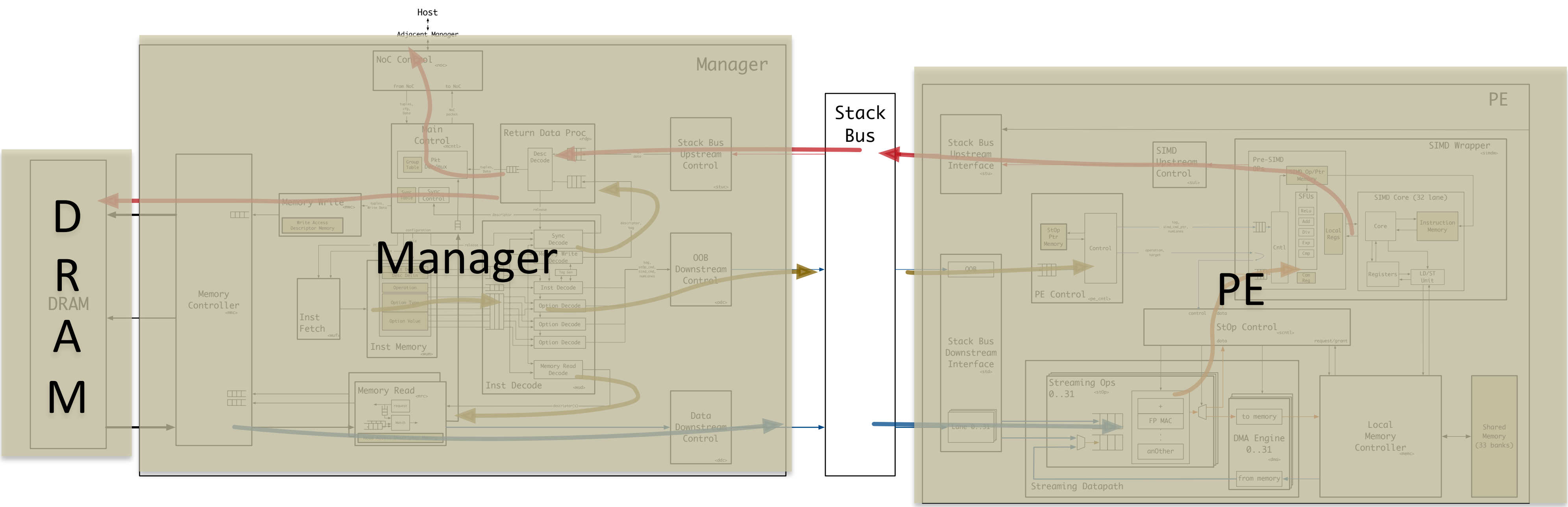
MAC or
Multiply

Processing (PE) Layer

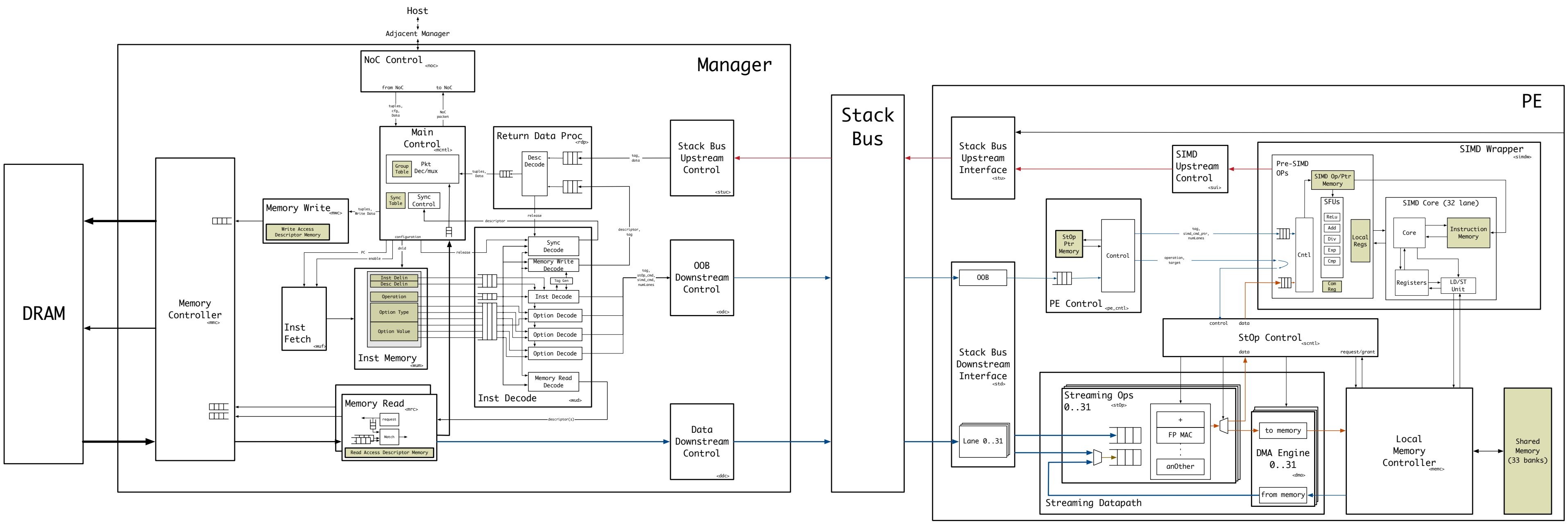
- Streaming operations operate directly on data
- SIMD Wrapper performs post stOp tasks



Detailed Block Diagram



Detailed Block Diagram



Test Performance

- Tested against multiple fan-ins

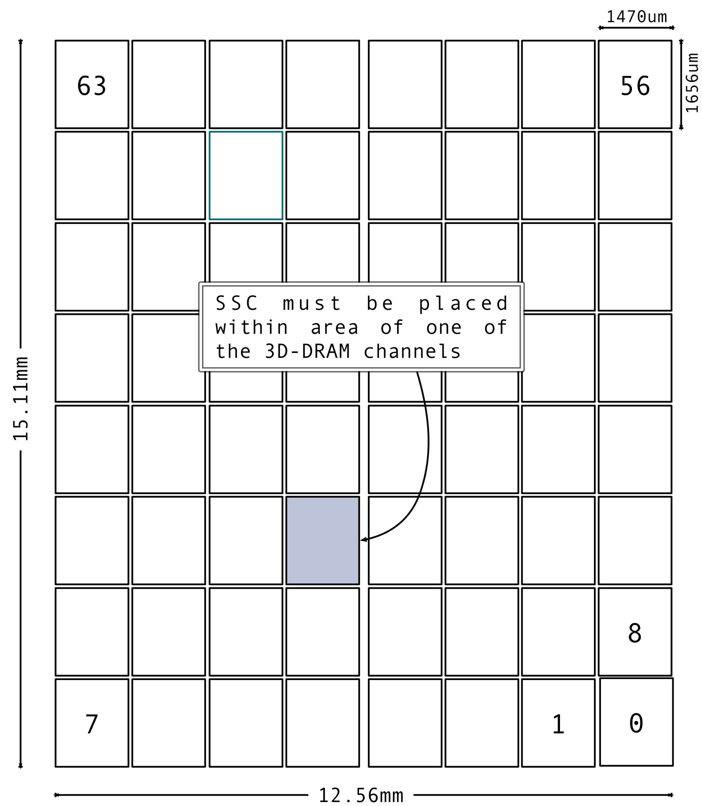
Test	System bandwidth (Tbps)
CONV2 [27]	25
CONV-294	26
CONV-300	27
CONV-500	29
CONV-1000	31
CONV-2500	32
FC-350	28
FC-500	29
FC-1000	31
FC-7 [27]	32

Baseline ANN Performance

Layer	fanin	Equivalent test	Percentage of instructions	Expected data bandwidth
1	363	CONV-300	44%	11.7
2	4	previous layer		
3	2400	CONV-2500	28%	9.1
4	4	previous layer		
5	2304	CONV-1000	10%	3.0
6	3456	CONV-2500	10%	3.2
7	3456	CONV-2500	7%	2.1
8	43264	FC-7	1%	0.2
9	4096	FC-7	1%	0.3
10	4096	FC-7	0.16%	0.1
			Total	29.7

AREA

- Area based on DiRAM4
  - SSC needs to fit into footprint of DiRAM4 sub-memory
  - available area is 2.43 sq-mm



Power/Area Scaling assumptions

- Scaling based on synthesizing representative block at 28nm and 65nm
  - Designed using 65nm library, 28nm is target technology
- Area

Area Scaling 65nm -> 28nm	
Memory	2.79
Logic	2.68

- Power

	Scaling Ratio		
	65nm to 28nm		
	Internal	Net switch	Leakage
Logic	5.07	1.21	4.12E-04
memory	5.07		

Manager AREA

- **Blocks**

	Manager		
	Distribution	65nm (mm)	28nm (mm)
Memory Controller	20.6%	0.99	0.36
NoC	6.9%	0.34	0.12
Read Control	47.1%	2.27	0.83
Write Control	6.7%	0.32	0.12
Instruction Proc	1.7%	0.08	0.03
Read data proc	1.6%	0.08	0.03
System Controller	1.6%	0.08	0.03
TSV	6.9%	0.33	0.33
Misc	6.8%	0.33	0.12
	100.0%	4.82	1.98

- **Utilization**

	Manager	
	65nm	28nm
Area used	4.82	1.98
Area available	6.66	2.43
Utilization	72%	81%
Utilization w/o TSV	71%	78%

PE AREA

- **Blocks**

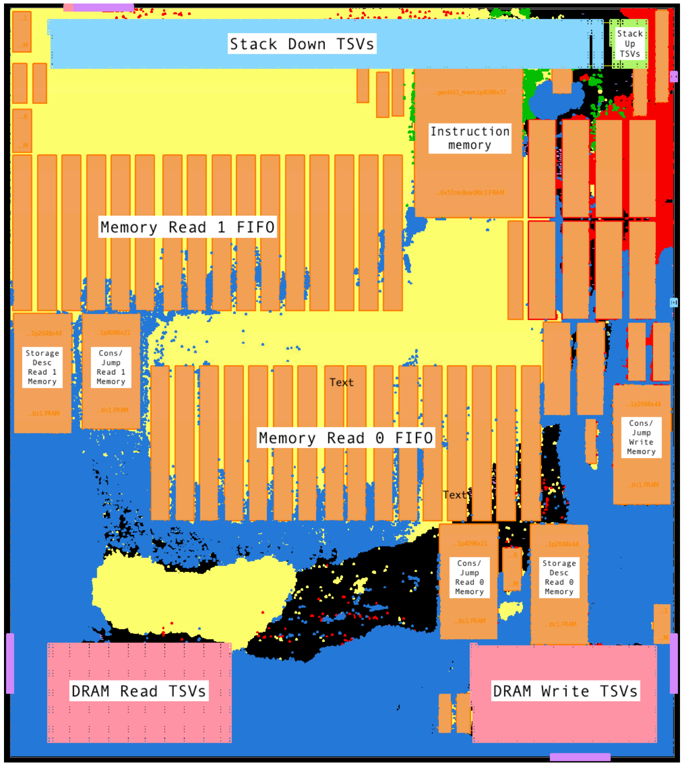
	Processing Engine		
	Distribution	65nm (mm)	28nm (mm)
Operation Decode	3.1%	0.11	0.04
Return Data Control	1.5%	0.05	0.02
SIMD Wrapper	15.1%	0.51	0.19
SIMD	17.9%	0.60	0.22
Streaming Ops	40.0%	1.34	0.49
StOp Control	1.9%	0.06	0.02
LM	16.4%	0.55	0.20
TSV	3.7%	0.12	0.12
Misc	0.4%	0.02	0.01
Total	100.0%	3.36	1.31

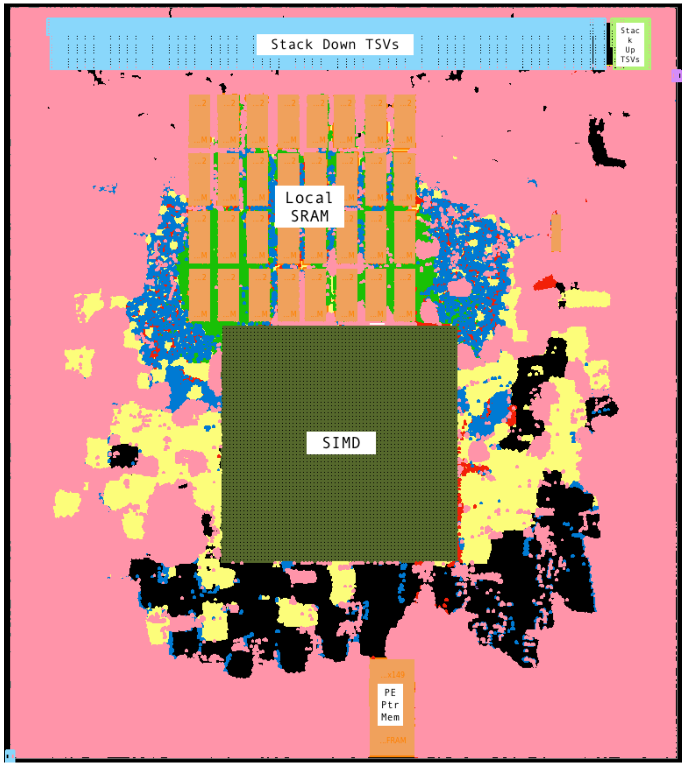
- **Utilization**

	Processing Engine	
	65nm	28nm
Area used	3.36	1.31
Area available	6.66	2.43
Utilization	50%	54%
Utilization w/o TSV	49%	51%

Area placement study

- Placed and routed without DRC or LVS
- Routing congestion showed minimal hotspots
- Parasitics used in Primetime for power analysis





Power Summary

- Power estimates based on CONV-294 testcase simulation
  - back to back operations
  - accumulate DRAM access and use DiRAM4 datasheet [14]
  - TSV capacitance from [4]
  - parasitics from layout

Parameters	
Frequency	500MHz
Test	CONV-294

  - designed using 65nm library
  - 28nm is target technology
  - power and area scaling based on synthesizing representative blocks

Block	Power (W)
Manager	42.55
PE	26.50
DRAM	4.51
DRAM TSVs	1.14
Stack Bus TSVs	0.74
Total	**75.44**

Comparison

- Best comparison is to NeuroStream and DaDianNao
  - Scaled

	TPU	NeuroStream	DaDianao	GPU
Power at capacity	38	83	3327	117
Ratio	50%	110%	4436%	156%
Power at bandwidth	3611	1117	167	1128
Ratio	4815%	1490%	223%	1505%
Power with bandwidth and capacity	3611	1117	3327	1128
Ratio	4815%	1490%	4436%	1505%
Area with bandwidth and capacity	27083	10055	14003	2532
Ratio	7738%	2873%	4001%	724%

  - Goldilocks ratio

	TPU	NeuroStream	DaDianao	GPU	this work
BW Utilization at capacity (Goldilocks ratio)	9028%	1270%	5%	451%	40%
	Coldest	Colder	Hot	Cold	Just right

Summary

- Real world applications will require multiple artificial neural networks
  - current solutions consume significant power and real-estate
- A 3DIC solution that includes:
  - A proposed custom 3D-DRAM
  - Instructions and data structures
  - Purpose designed functions to accelerate a Artificial Neural Networks
  - Architecture to take advantage of 3D technology
- Overall area and power improvement

Publication(s)

- Multi-ANN Edge System based on a Custom 3DIC DRAM - submitted to IEEE journal on emerging technologies and selected topics
- Using a 2-D Laser Scanner along with Cogent Confabulation as a Localized Navigation Aid
 - still trying to find an appropriate journal

References

BIBLIOGRAPHY

[1] Abadi, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[2] Aizenberg, I. et al. *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Springer Science & Business Media, 2013.

[3] Azarkhish, E. et al. "Neurostream: Scalable and Energy Efficient Deep Learning with Smart Memory Cubes". *arXiv preprint arXiv:1701.06420* (2017).

[4] Bamberg, L. & Garcia-Ortiz, A. "High-Level Energy Estimation for Submicrometric TSV Arrays". *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* **25**.10 (2017), pp. 2856–2866.

[5] Brette, R. "Philosophy of the Spike: Rate-Based vs. Spike-Based Theories of the Brain". *Frontiers in Systems Neuroscience* **9** (2015), p. 151.

[6] Brunel, N. & Rossum, M. C. W. van. "Lapicque's 1907 paper: from frogs to integrate-and-fire". *Biological Cybernetics* **97**.5 (2007), pp. 337–339.

[7] Bullinaria, D. J. A. *Introduction to Neural Computation : Neural Computation*. `http://www.cs.bham.ac.uk/~jxb/inc.html`. Accessed: 2017-09-08.

[8] Carnevale, N. & Hines, M. *The NEURON Book*. Cambridge University Press, 2006.

[9] Chen, T. et al. "Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning". *ACM Sigplan Notices*. Vol. 49. 4. ACM. 2014, pp. 269–284.

[10] Chen, Y. et al. "DaDianNao: A Machine-Learning Supercomputer". *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 2014, pp. 609–622.

[11] Chen, Y.-H. et al. "14.5 Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks". *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE. 2016, pp. 262–263.

[12] Chen, Y. et al. "DianNao Family: Energy-efficient Hardware Accelerators for Machine Learning". *Commun. ACM* **59**.11 (2016), pp. 105–112.

[13] *DDR3L SDRAM*. D 8/17 EN. Micron Technology Inc. 2015.

[14] *DiRAM4-64Cxx Cached Memory Subsystem*. Rev. 0.04. Tezzaron Semiconductor. 2015.

[15] Esmaeilzadeh, H. et al. "NnSP: embedded neural networks stream processor". *48th Midwest Symposium on Circuits and Systems, 2005*. IEEE. 2005, pp. 223–226.

[16] *Evolving 2.5D and 3D Integration*. Tezzaron Semiconductor.

[17] Farabet, C. et al. "Neuflow: A runtime reconfigurable dataflow processor for vision". *Cvpr 2011 Workshops*. IEEE. 2011, pp. 109–116.

[18] Hinton, G. E. et al. "A Fast Learning Algorithm for Deep Belief Nets". *Neural Computation* **18**.7 (2006). PMID: 16764513, pp. 1527–1554. eprint: `https://doi.org/10.1162/neco.2006.18.7.1527`.

[19] Hochreiter, S. & Schmidhuber, J. "Long short-term memory". *Neural computation* **9**.8 (1997), pp. 1735–1780.

[20] ITRS. *International Technology Roadmap for Semiconductors 2.0, Interconnect*. 2015.

[21] Izhikevich, E. M. "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks* **15**.5 (2004), pp. 1063–1070.

[22] Jacob, B. et al. *Memory Systems: Cache, DRAM, Disk*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[23] Jouppi, N. P. et al. "In-datacenter performance analysis of a tensor processing unit". *arXiv preprint arXiv:1704.04760* (2017).

[24] Kim, D. et al. "Neurocube: A Programmable Digital Neuromorphic Architecture with High-Density 3D Memory". *Proceedings of ISCA*. Vol. 43. 2016.

[25] Kim, S. W. et al. "Ultra-Fine Pitch 3D Integration Using Face-to-Face Hybrid Wafer Bonding Combined with a Via-Middle Through-Silicon-Via Process". *2016 IEEE 66th Electronic Components and Technology Conference (ECTC)*. 2016, pp. 1179–1185.

[26] Krizhevsky, A. et al. *ImageNet Classification with Deep Convolutional Neural Networks*. `http://image-net.org/challenges/LSVRC/2012/supervision.pdf`. Accessed: 2016-08-30.

[27] Krizhevsky, A. et al. "Imagenet classification with deep convolutional neural networks". *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[28] Kwolek, B. "Face Detection Using Convolutional Neural Networks and Gabor Filters". *Artificial Neural Networks: Biological Inspirations – ICANN 2005: 15th International Conference, Warsaw, Poland, September 11-15, 2005. Proceedings, Part I*. Ed. by Duch, W. et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 551–556.

[29] Le, Q. V. "Building high-level features using large scale unsupervised learning". *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 8595–8598.

[30] Liu, Y. et al. "A compact low-power 3D I/O in 45nm CMOS". *2012 IEEE International Solid-State Circuits Conference*. IEEE. 2012, pp. 142–144.

References

[31] Luo, T. et al. "DaDianNao: A Neural Network Supercomputer". *IEEE Transactions on Computers* **66**.1 (2017), pp. 73–88.

[32] Maas, A. L. et al. "Rectifier nonlinearities improve neural network acoustic models". *Proc. ICML*. Vol. 30. 1. 2013.

[33] Maddison, C. J. et al. "Move evaluation in go using deep convolutional neural networks". *arXiv preprint arXiv:1412.6564* (2014).

[34] Nielsen, M. *NNs and DL*. `http://www.neuralnetworksanddeeplearning.com/index.html`. Accessed: 2018-01-02.

[35] Nvidia®. *NVidia Tesla P100 Datasheet*. `http://images.nvidia.com/content/tesla/pdf/nvidia-tesla-p100-datasheet.pdf`. Accessed: 2017-12-29.

[36] Patti, R. "2.5 D and 3D Integration Technology Update". *Additional Papers and Presentations* **2014**.DPC (2014), pp. 1–35.

[37] Paugam-Moisy, H. & Bohte, S. "Computing with spiking neuron networks". *Handbook of natural computing*. Springer, 2012, pp. 335–376.

[38] Qiu, Q. et al. "A parallel neuromorphic text recognition system and its implementation on a heterogeneous high-performance computing cluster". *IEEE Transactions on Computers* **62**.5 (2013), pp. 886–899.

[39] Schabel, J. C. *Design of an Application-Specific Instruction Set Processor for the Sparse Neural Network Design Space*. North Carolina State University. Box 7911, Raleigh, NC 27695-7911, 2017.

[40] Schabel, J. C. et al. *Predictive energy-Per-Op scaling for exploring the design space*. North Carolina State University. Box 7911, Raleigh, NC 27695-7911, 2014.

[41] Standard, D. S. *JEDEC JESD79-3*. 2007.

[42] Taigman, Y. et al. "DeepFace: Closing the Gap to Human-Level Performance in Face Verification". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014.

[43] Team, D. D. *Introduction to Deep Neural Networks*. `http://deeplearning4j.org`. Accessed: 2018-01-02.

[44] Various. *Backpropagation*. `https://en.wikipedia.org/wiki/Backpropagation`. Accessed: 2018-01-02.

[45] Various. *Neuron*. `https://en.wikipedia.org/wiki/Neuron`. Accessed: 2018-01-02.

[46] Various. *Neuron*. `https://en.wikipedia.org/wiki/Deep_learning`. Accessed: 2018-01-02.

[47] Various. *Pooling Overview*. `http://ufldl.stanford.edu/tutorial/supervised/Pooling/`. Accessed: 2018-02-16.

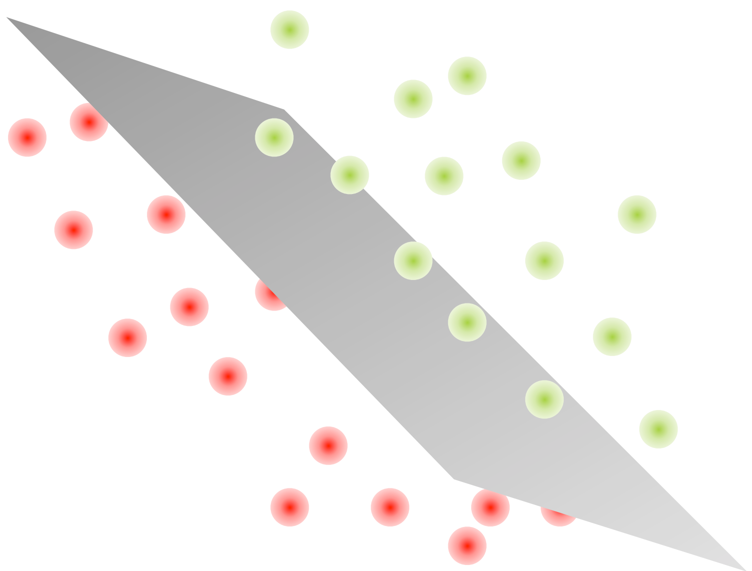
[48] Various. *Softmax function*. `https://en.wikipedia.org/wiki/Softmax_function`. Accessed: 2018-01-02.

[49] Various. *Softmax regression*. `http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/`. Accessed: 2018-02-16.

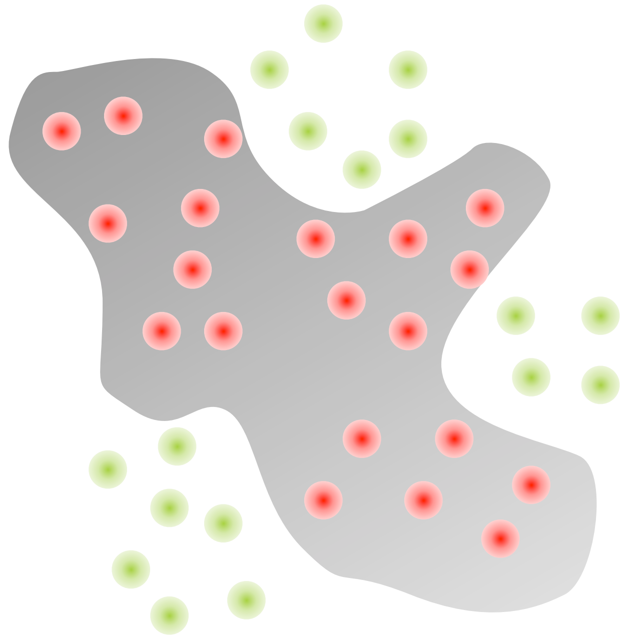
[50] Various. *Stochastic gradient descent*. `https://en.wikipedia.org/wiki/Stochastic_gradient_descent`. Accessed: 2018-01-02.

Backup

Discrimination



Linear

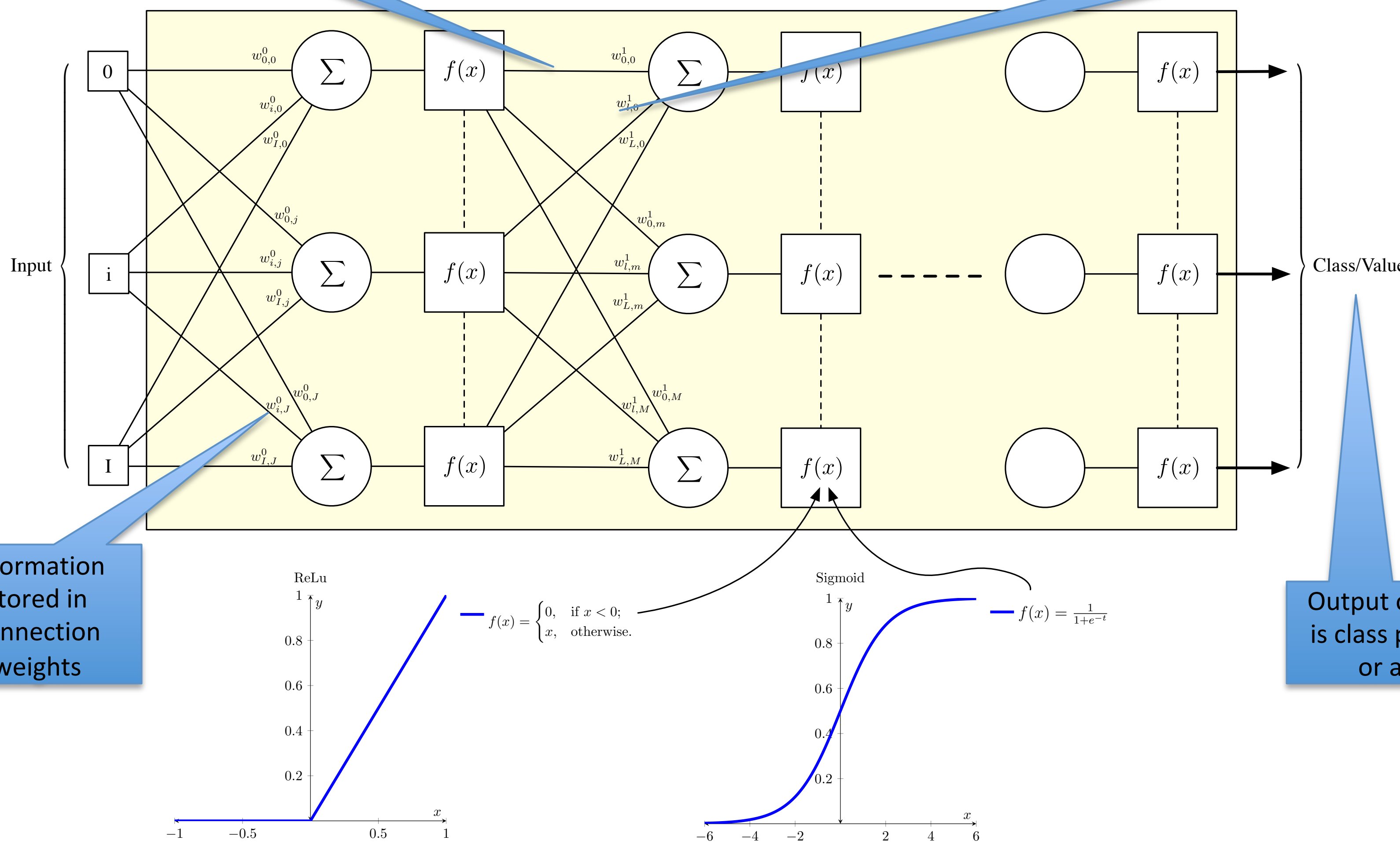


Complex

Artificial Neural Networks
Rate-based

Output of neuron is fixed value

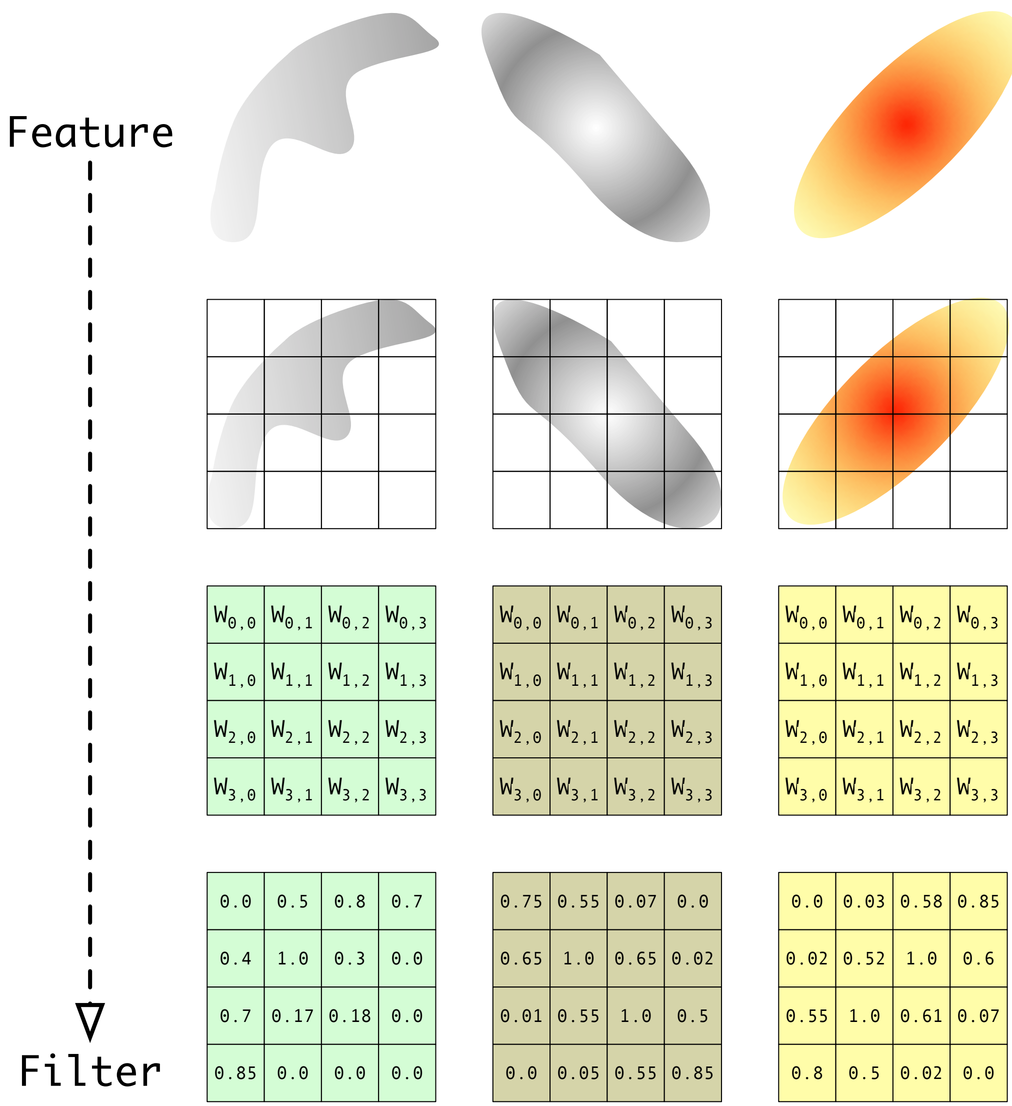
Multiple layers of neurons
Outputs of each layer feeds next layer



Information stored in connection weights

Output of network is class probability or a value

Feature kernels

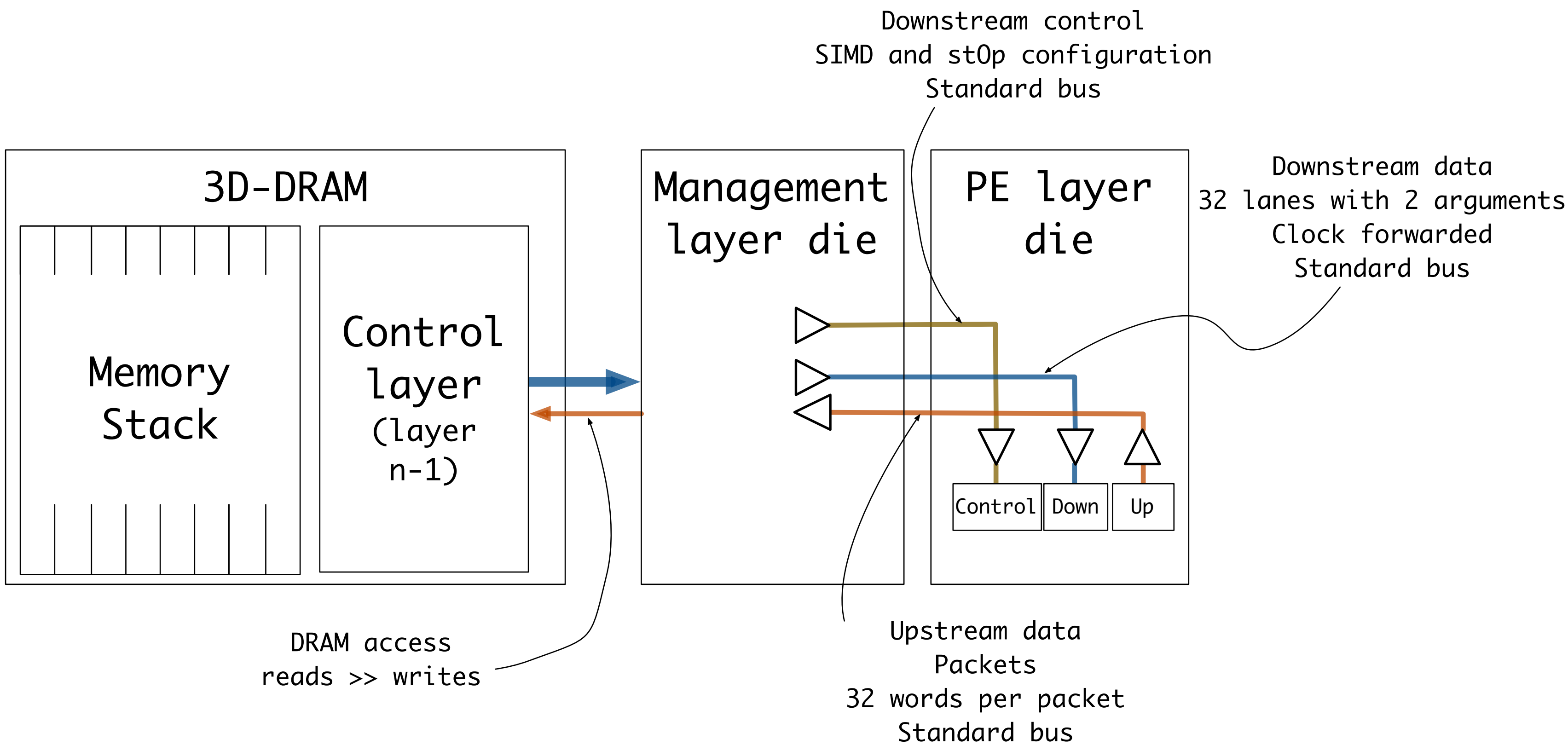


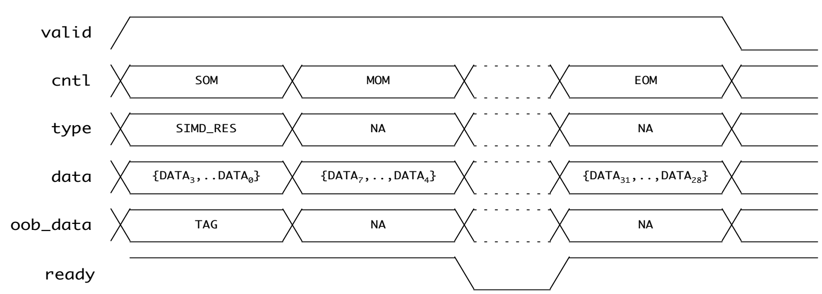
Power/Area Scaling assumptions

- Scaling based on synthesizing representative block at 28nm and 65nm
  - designed using 65nm library
  - 28nm is target technology

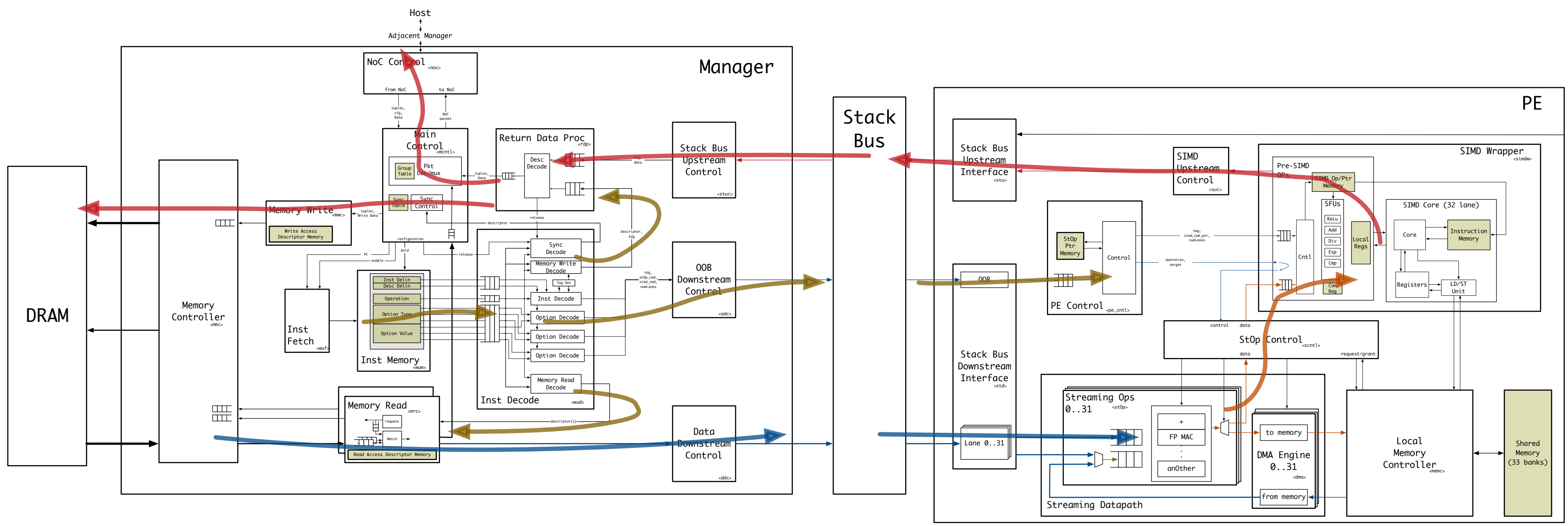
	Synthesis numbers					
	65nm			28nm		
	Internal	Net switch	Leakage	Internal	Net switch	Leakage
Logic	66.9	1.53	2.02	13.2	1.26	4900
memory	2.36			0.0438		

DRAM and Stack Buses



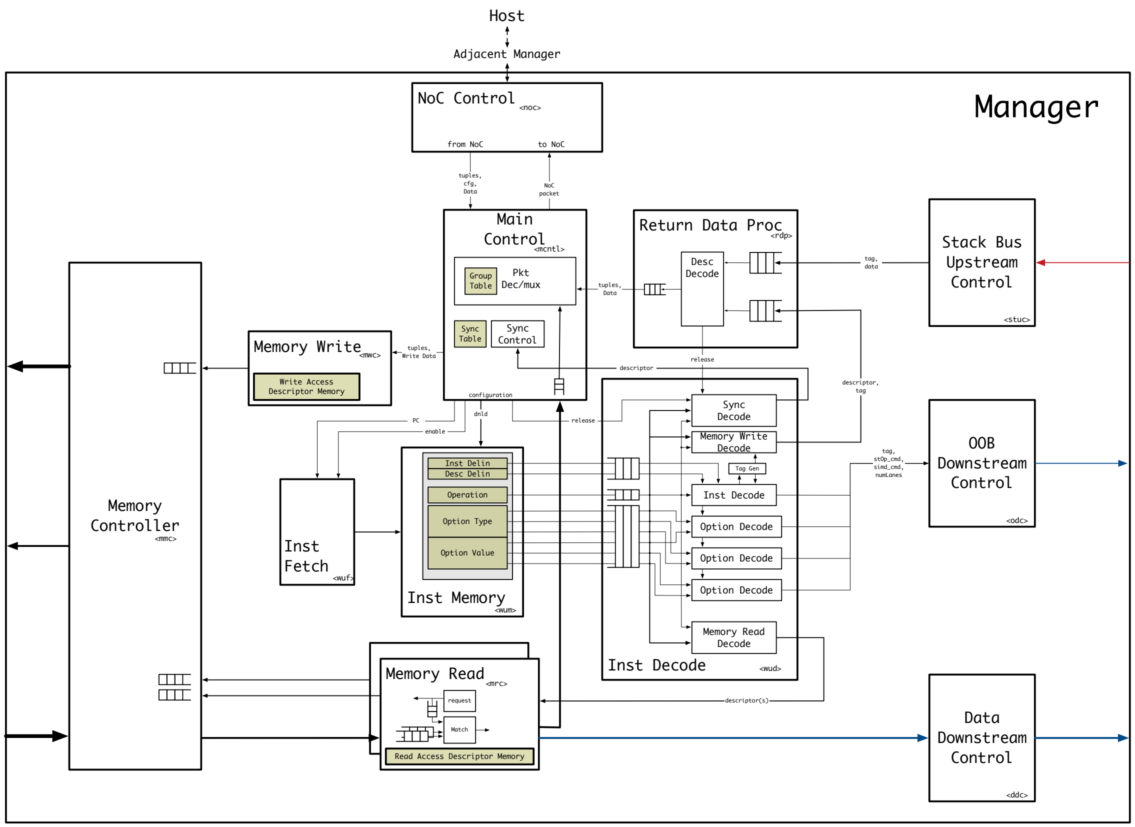


Detailed Block Diagram

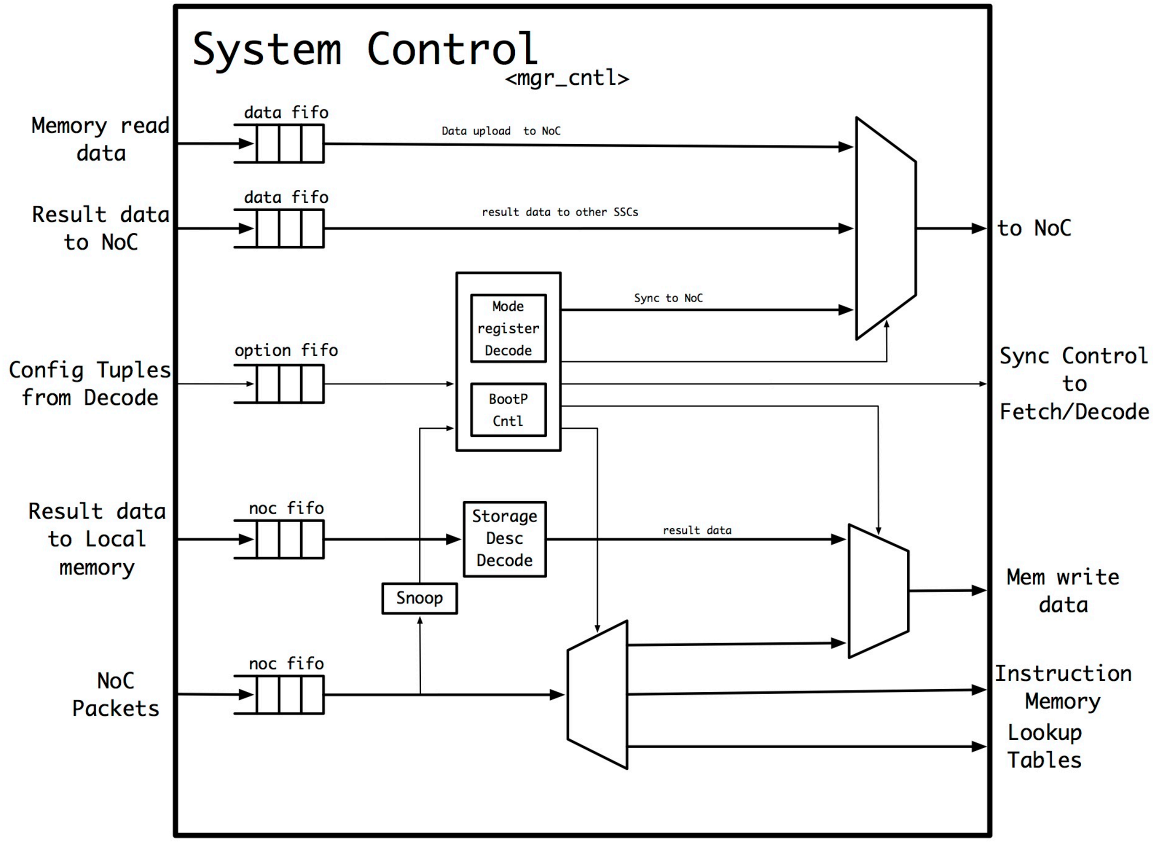


Manager Layer

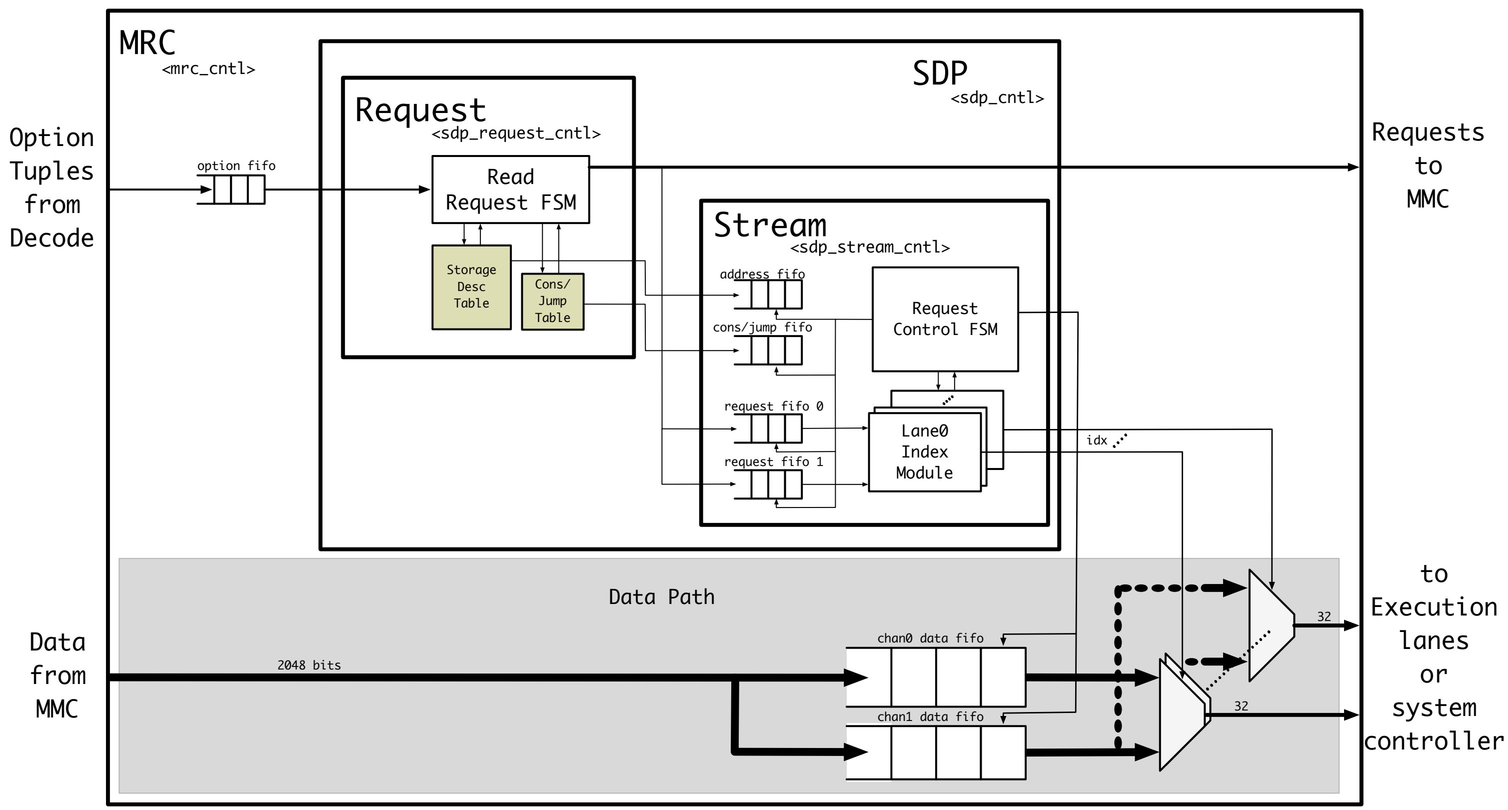
- Decodes instructions
  - sub-descriptors are sent to dependent blocks
- Reads and writes to main memory
- Communicates to host and other SSCs



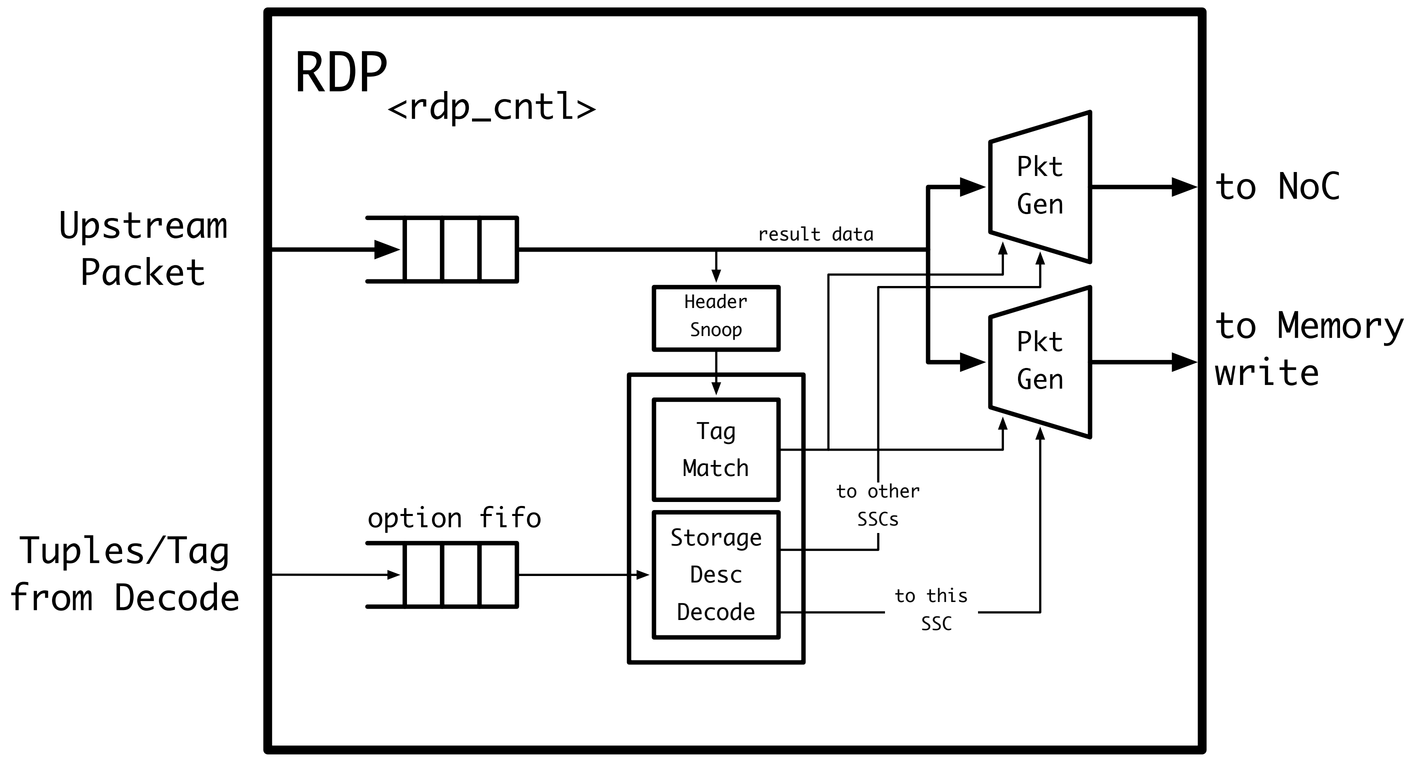
Manager Controller



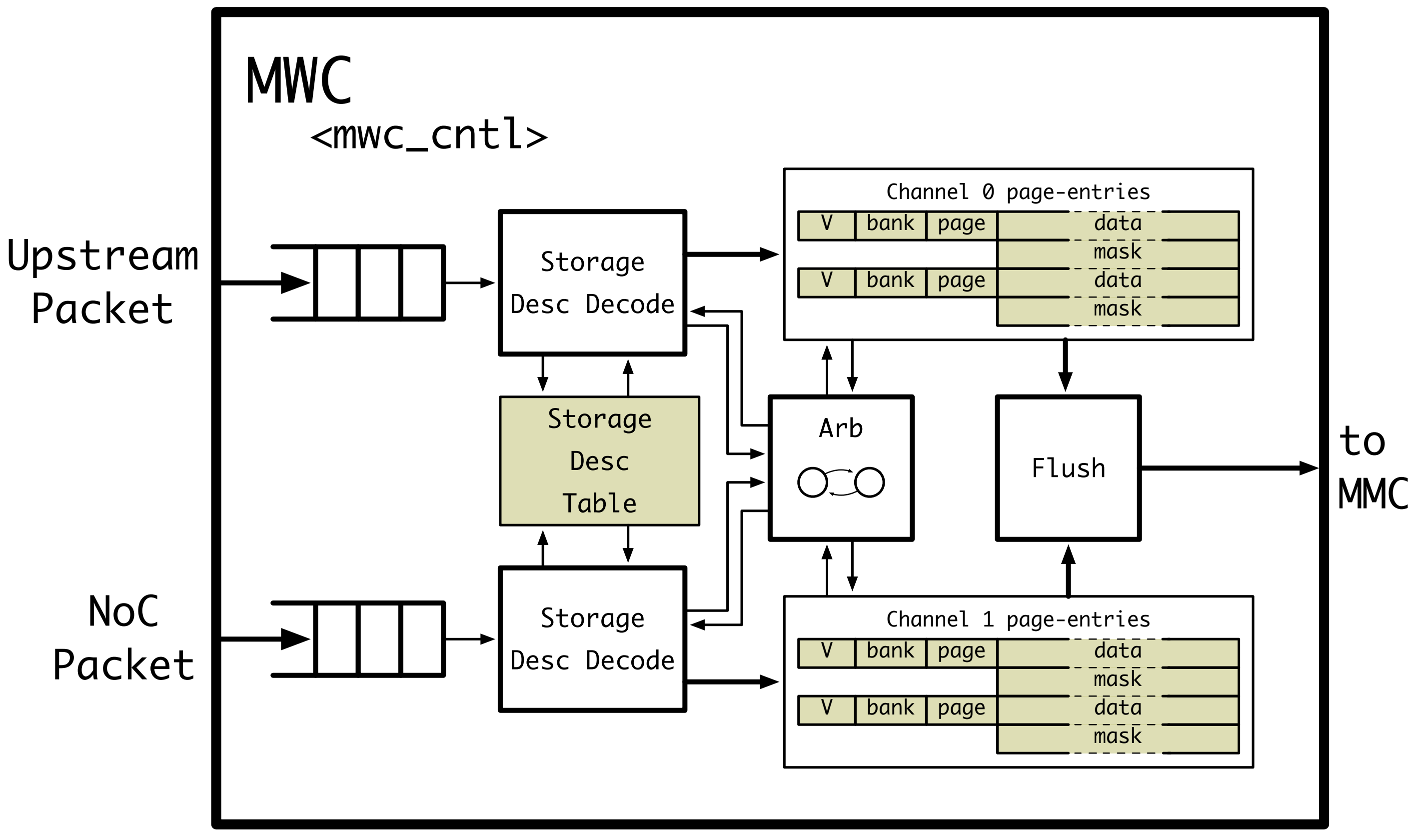
MRC


Option
Tuples
from
Decode
MRC
<mrc_cntl>
option fifo
SDP
<sdp_cntl>
Request
<sdp_request_cntl>
Read
Request FSM
Storage
Desc
Table
Cons/
Jump
Table
Stream
<sdp_stream_cntl>
address fifo
cons/jump fifo
Request
Control FSM
request fifo 0
request fifo 1
Lane0
Index
Module
idx
Requests
to
MMC
Data Path
Data
from
MMC
2048 bits
chan0 data fifo
chan1 data fifo
32
32
to
Execution
lanes
or
system
controller

RDP


RDP
<rdp_cntl>
Upstream
Packet
Tuples/Tag
from Decode
option fifo
result data
Header
Snoop
Tag
Match
Storage
Desc
Decode
to other
SSCs
to this
SSC
Pkt
Gen
Pkt
Gen
to NoC
to Memory
write

MWC

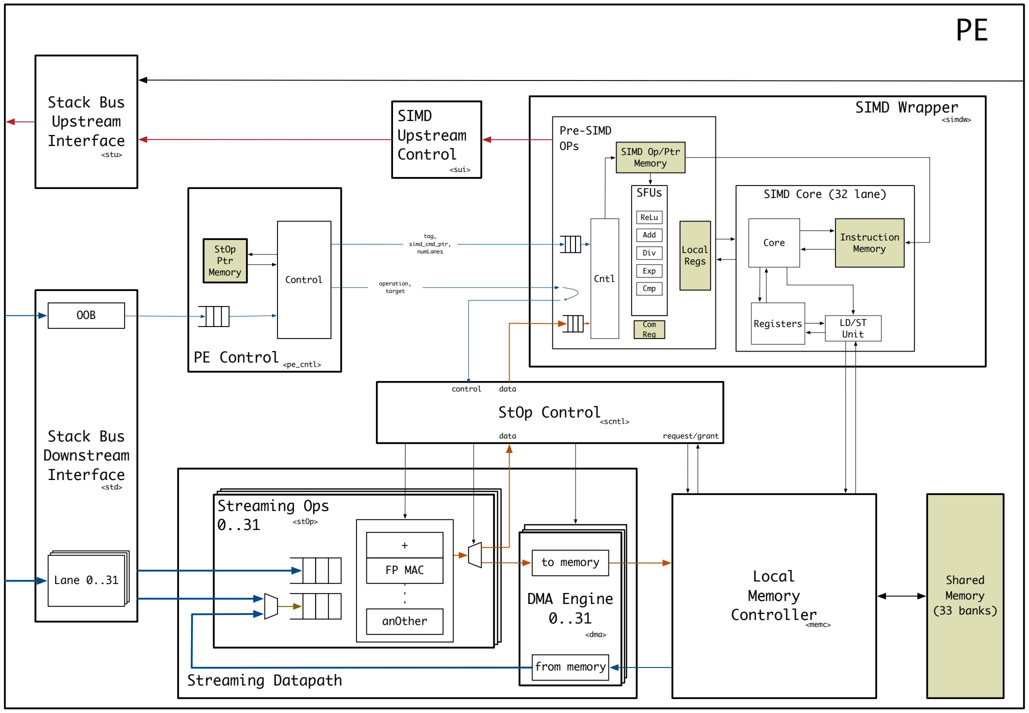


Upstream
Packet
NoC
Packet
MWC
<mwc_cntl>
Storage
Desc Decode
Storage
Desc
Table
Storage
Desc Decode
Channel 0 page-entries
V
bank
page
data
mask
data
mask
Arb
Flush
Channel 1 page-entries
V
bank
page
data
mask
data
mask
to
MMC

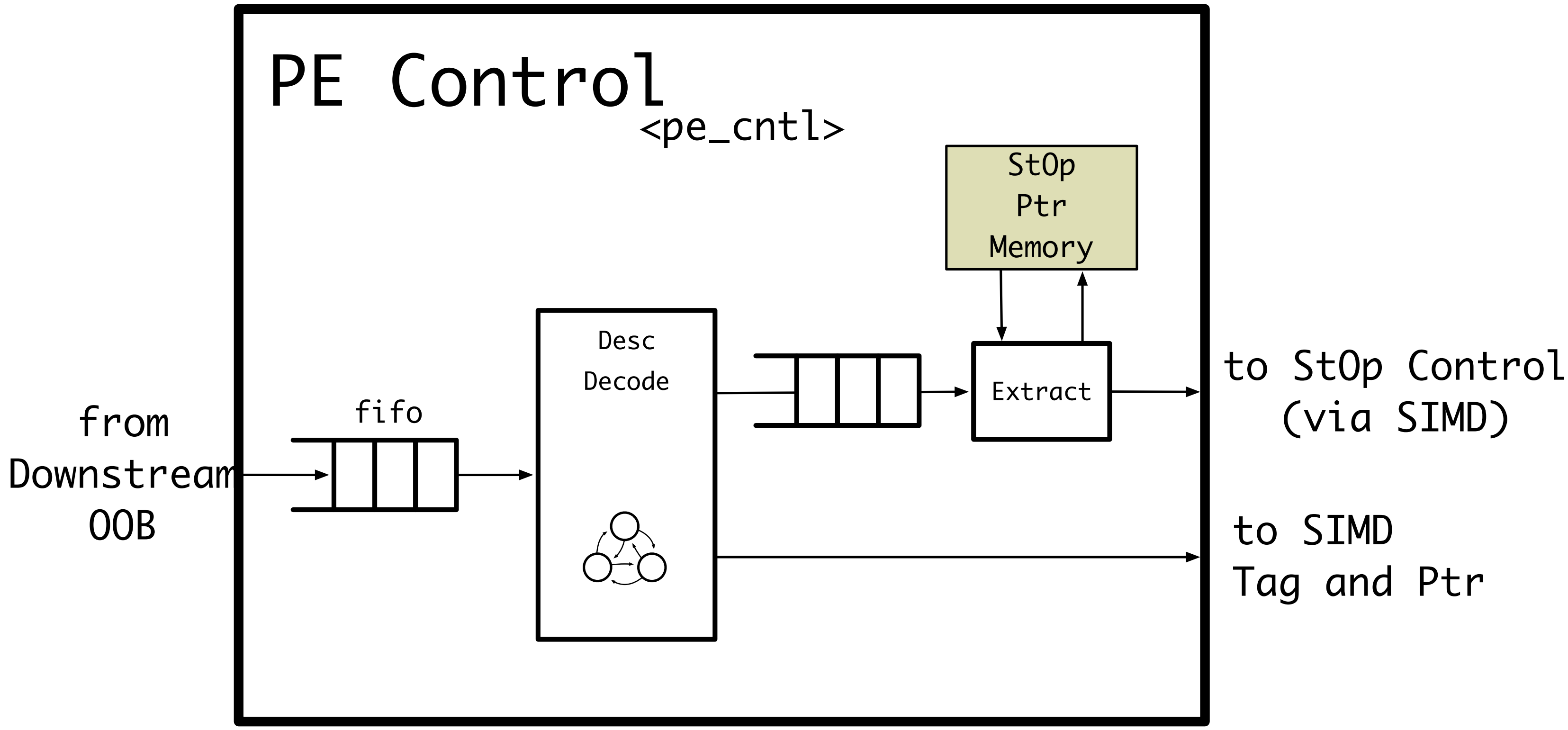
NC STATE UNIVERSITY

Processing (PE) Layer

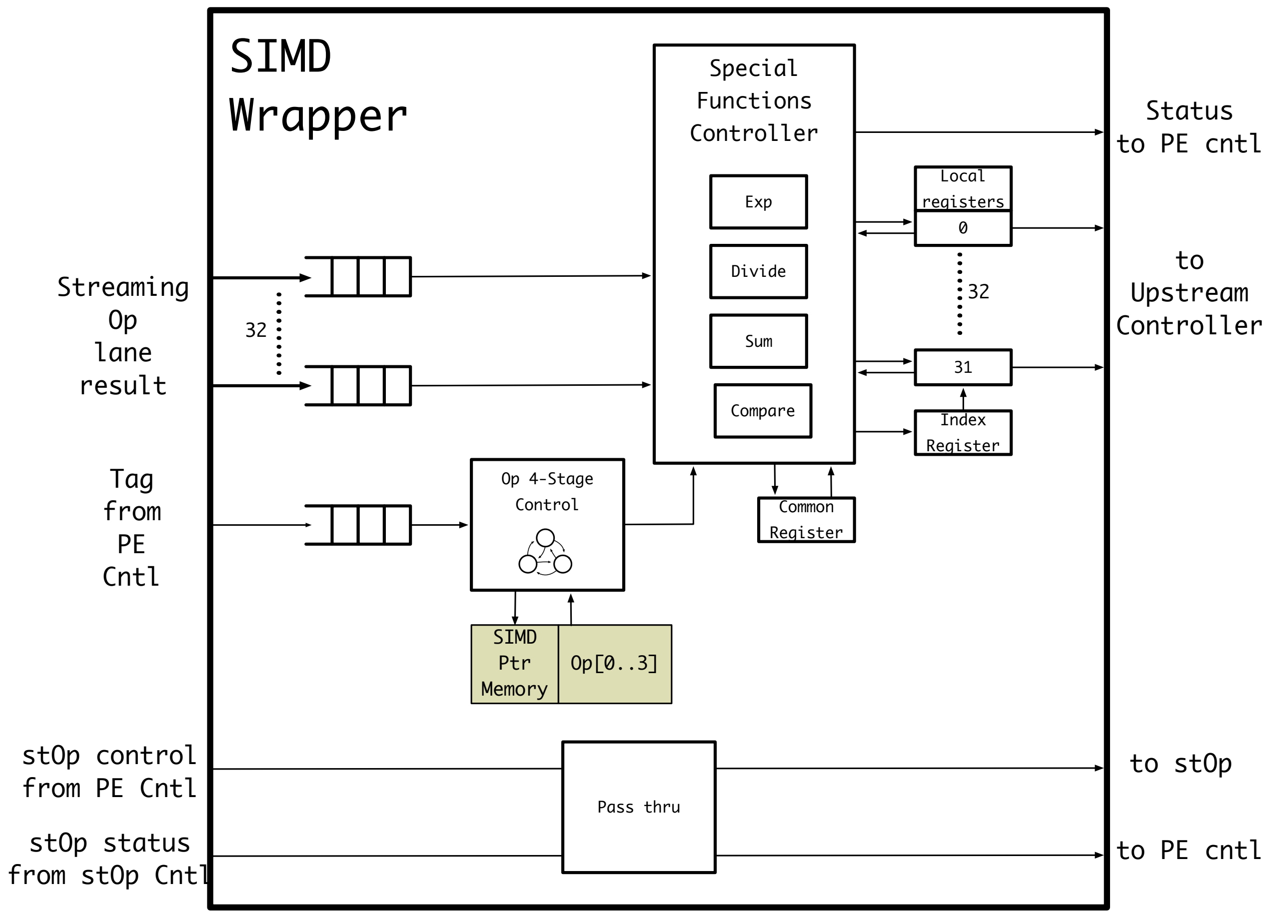
- Streaming operations operate on data directly from the Stack bus
  - MAC, Multiply
- SIMD Wrapper performs post stOp tasks
  - add/divide/exponent, sends result back to manager



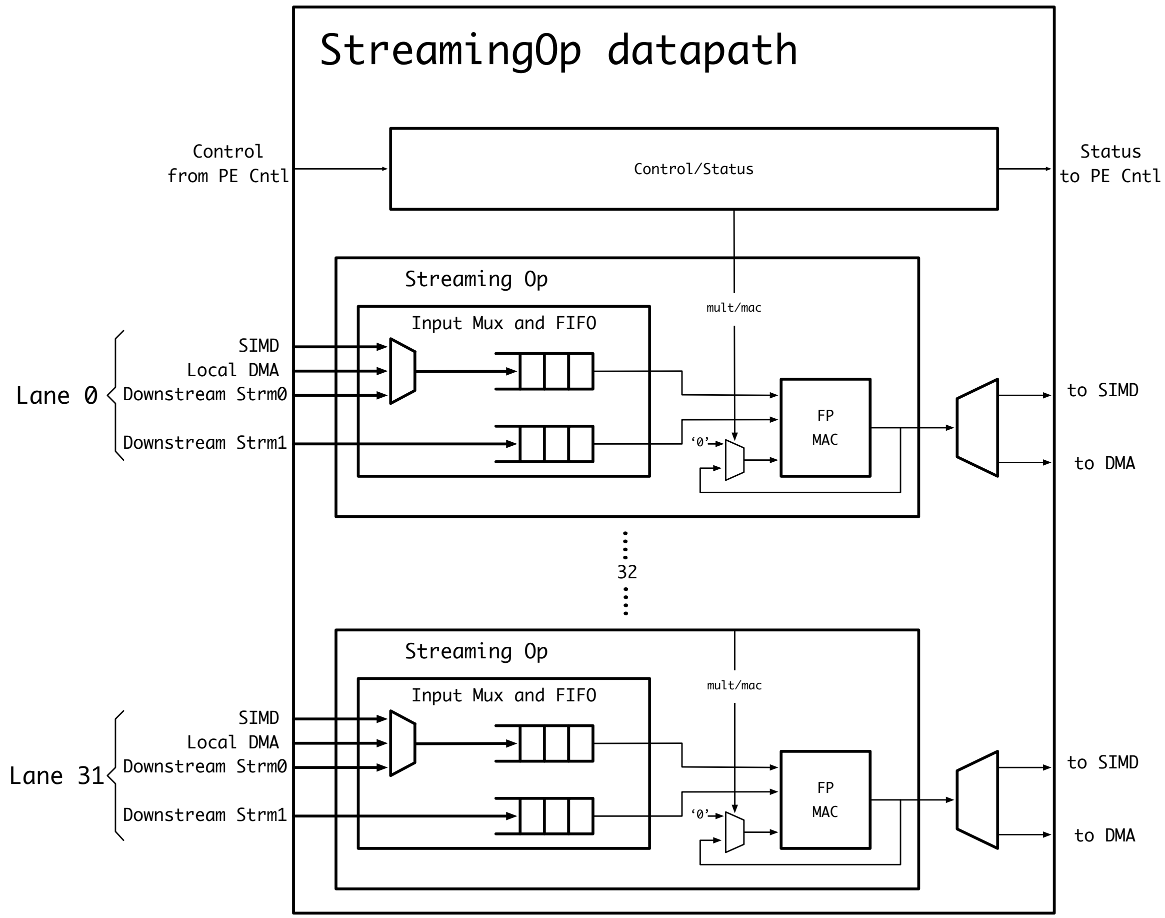
PE Controller



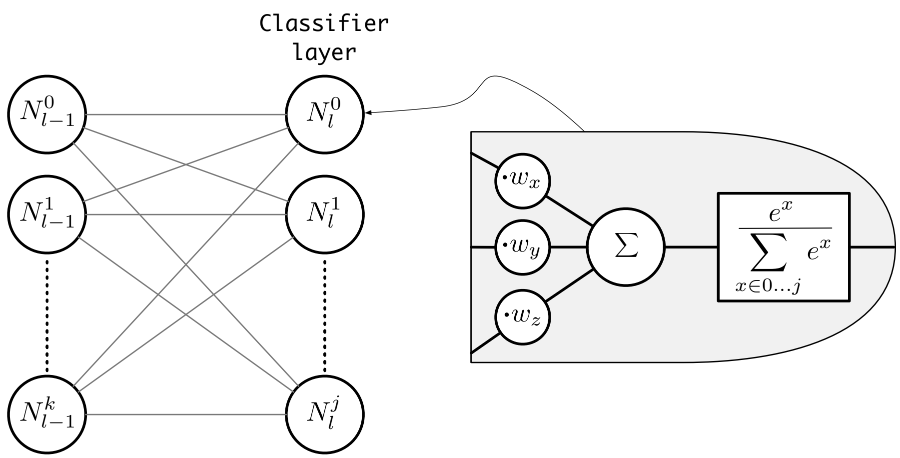
SIMD Wrapper



Streaming Operations

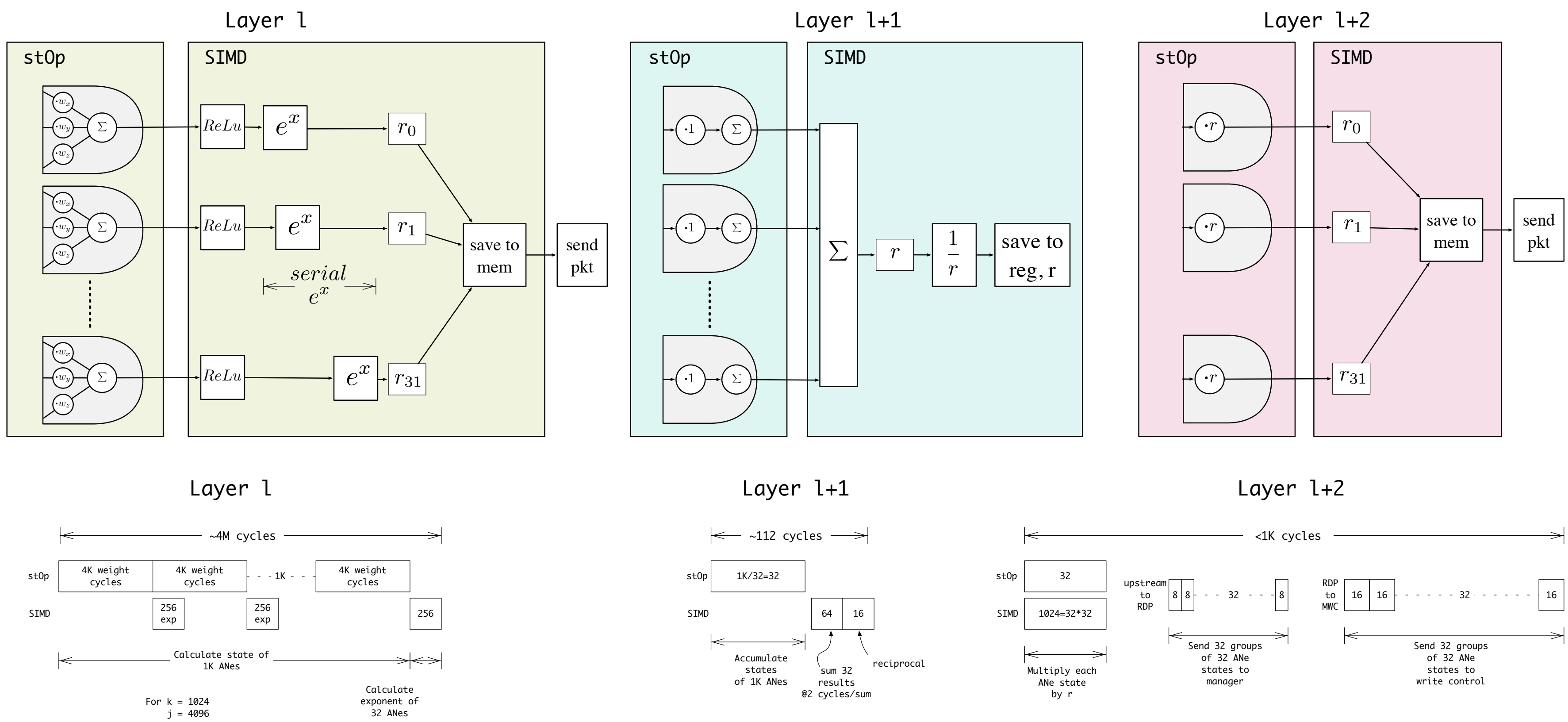


Classifier Implementation



- Separate into sub-layers

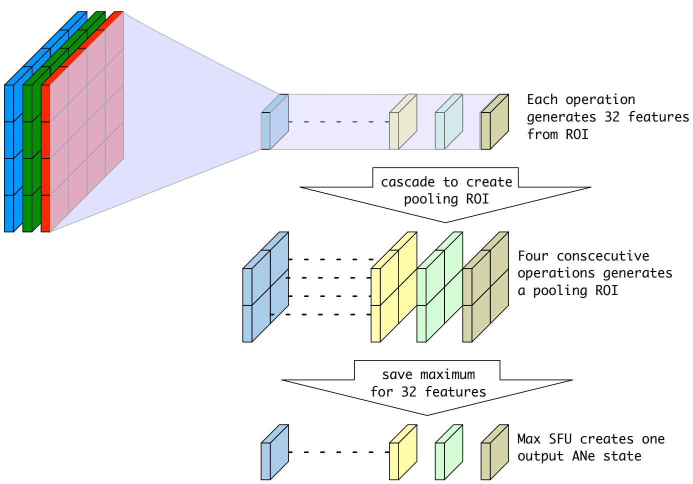
Classifier Implementation

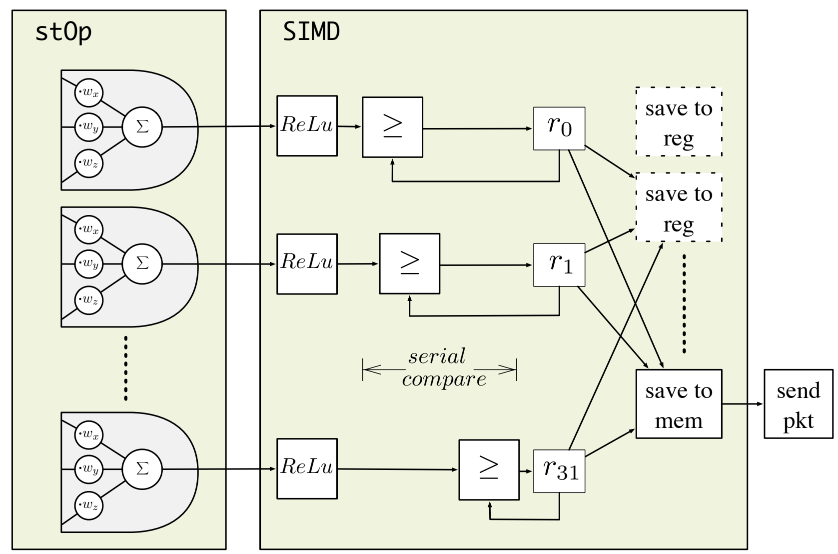


- Additional processing absorbed by layer l ANe MAC operation

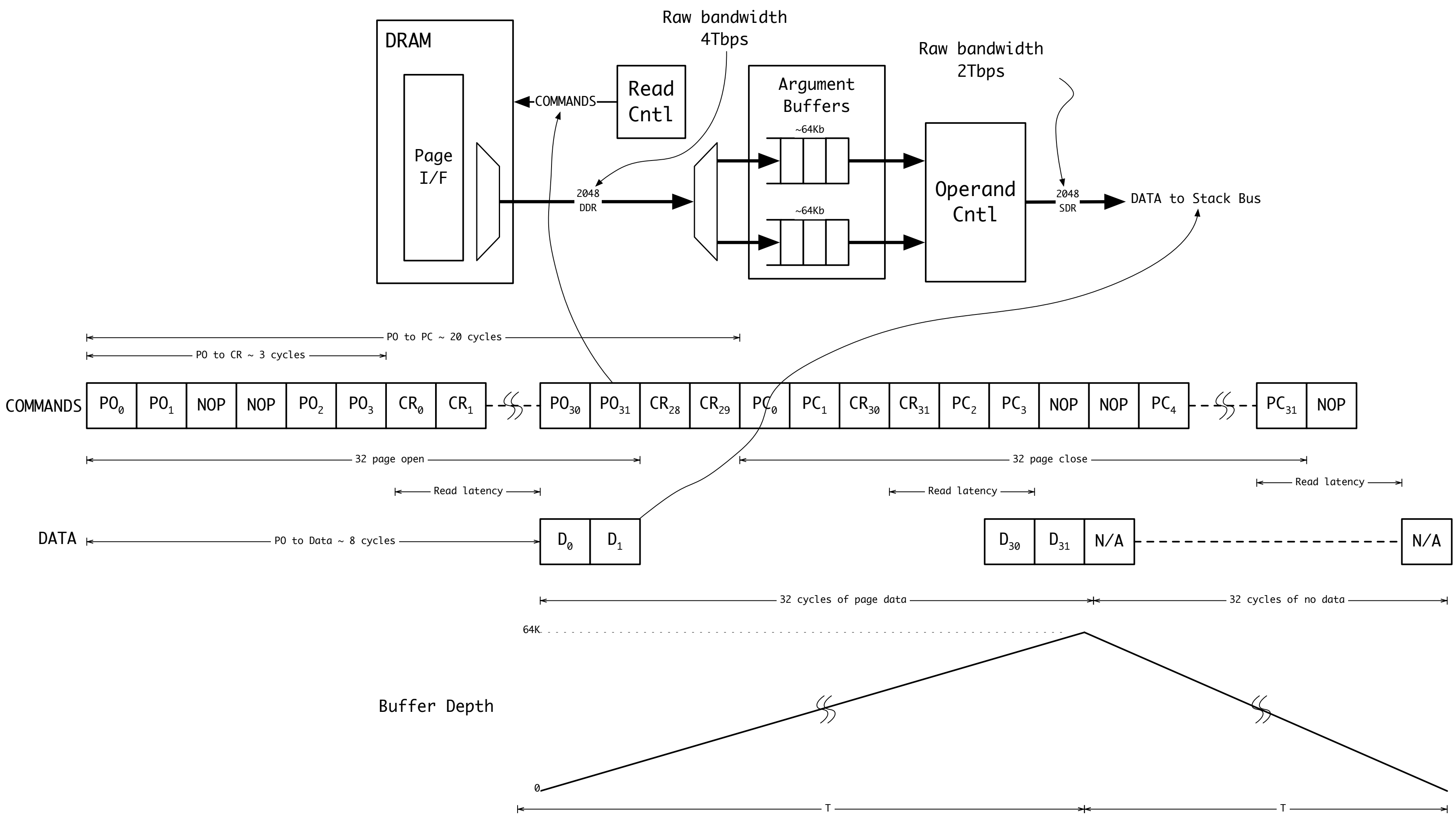
Pooling Implementation

- Pooling performed by ordering previous layers calculation





Example DRAM Access Sequence



State of the Art
Older ASIC's

- NeuroCube[1]
  - uses HMC 3D-DRAM but dependence on SRAM limits support to CNNs
- Eyeriss[2]
  - discusses DRAM accesses but their focus is on the convolution operation
  - supports CNN convolution operation only
- NnSP[3]
  - uses cache between DRAM and 8Mb SRAM and will be limited by DRAM bandwidth
  - does not discuss multi-devices
- Neuflow[4]
  - limits their support to CNNs and in the case of Eyeriss only supports the convolution
  - will not support locally-connected ANNs

[1] [Kim16], [2] [Che16], [3] [Esm05], [4] [Far11]

State of the Art
GPU's

- GPU's target the general market and employ infrastructures not necessary for NN implementation
  - we associate power of GPU solution ~100-200W
  - Will still be limited by memory bandwidth although they are starting to utilize 3D-DRAM
  - general purpose architecture makes it hard/impossible to make full use of theoretical performance

	CPU	V6	mGPU	IBM	GPU
Peak GOPs	10	160	182	1280	1350
Real GOPs	1.1	147	54	1164	294
Power W	30	10	30	5	220
GOPs/W	0.04	14.7	1.8	230	1.34

Table 5. Performance comparison. 1- CPU: Intel DuoCore, 2.7GHz, optimized C code, 2- V6: neuFlow on Xilinx Virtex 6 FPGA—on board power and GOPs measurements; 3- IBM: neuFlow on IBM 45nm process: simulated results, the design was fully placed and routed; 4- mGPU/GPU: two GPU implementations, a low power GT335m and a high-end GTX480.

[Far11]

Do not use backup

Example Image Recognition CNN

- Example taken from Krizhevsky, Sutsever, Hinton 2012 [Kri12]
- Seven layers but only first 3 layers shown
- ~60M parameters(~2Gb), ~2GFLOP

FMA Power/Area Estimate

- Consider an FMA running at the Stack bus speed of ~1GHz, from [GH11] table 1

Fused Multiply-Accumulate Power and Area [4]

Pipeline	Freq (GHz)	Area (um^2)	Static Power (mW)	Dynamic Power (mW)	$W/GFlop$	$mm^2/GFlop$	Power Density (W/mm^2)
6	2.08	16077	1.2	30.9	0.0077	0.0039	2.00
6	2.08	16077	1.2	30.9	0.0077	0.0039	2.00
5	1.32	14241	0.55	12.85	0.0051	0.0054	0.94
4	0.98	12670	0.58	8.09	0.0044	0.0065	0.68
3	0.5	12117	0.16	3.16	0.0033	0.0121	0.27
3	0.2	10619	0.0358	0.952	0.0025	0.0265	0.09

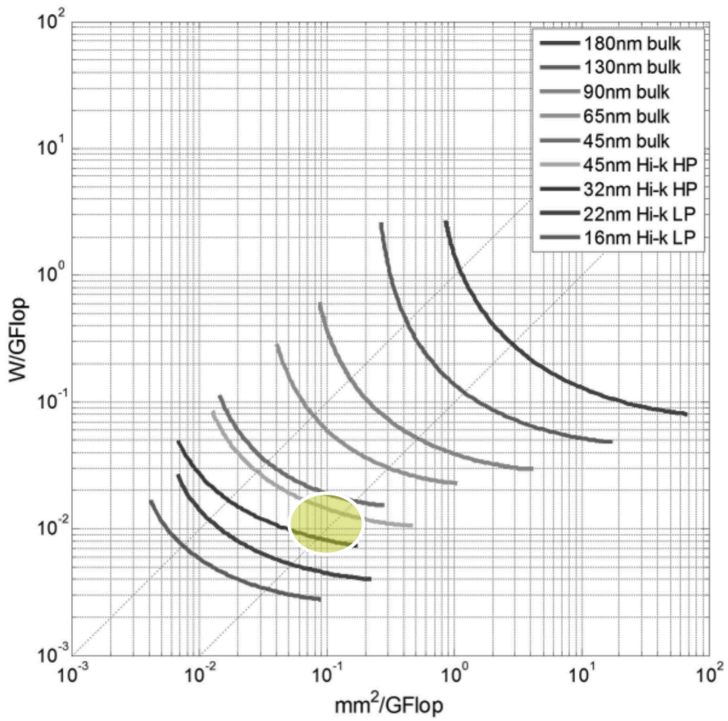
$$\begin{aligned}\text{FMA Area (45nm)} &= area/fma \cdot \#PE \cdot fma/PE \\ &= \frac{12670}{10^6} \cdot 64 \cdot 32 = 25.95\,\mathrm{mm}^2\end{aligned}$$

$$\begin{aligned}\text{Max FMA Power (45nm)} &= \text{Number of FMA} \cdot (\text{leakage power} + \text{dynamic Power}) \\ &= 64 \cdot 32 \cdot \left(0.58 + \frac{8.09}{0.98} \cdot 1.0\right) = 1.187 + 16.91 \\ &= 18.1\,\mathrm{W}\end{aligned}$$

$$\text{Actual FMA Power (45nm)} \approx \frac{54\text{Tbps}}{131\text{Tbps}} \cdot 16.91 + 1.187 = 6.97 + 1.187 = 8.16\,\mathrm{W}$$

FMA Power/Area Estimate

- From [GH11] figure 10 scaling from 45nm to 32nm whilst maintaining the mm^2/Gflop suggests a scaling factor of 0.53



$$\text{Actual FMA Power (32nm)} \approx \frac{0.8}{1.4} \cdot 8.16 = 4.66\,\text{W}$$

$$\text{FMA Area (32nm)} = 25.95\,\text{mm}^2$$

TSV Area

- Power Delivery
  - ~2000TSV's ~0.1mm^2
- Stack Bus
  - 2737 signals
  - Assume 2:1 for GND/VCC ~ 5500 TSV's
  - 5um pitch ~ 0.14mm^2/PE
  - 8.8mm^2 for 64 PE's
- DRAM Bus
  - 4255 signals ~ 8500 TSV's
  - 13.6mm^2 for 64 Ports

TSV Power

from [Liu12]

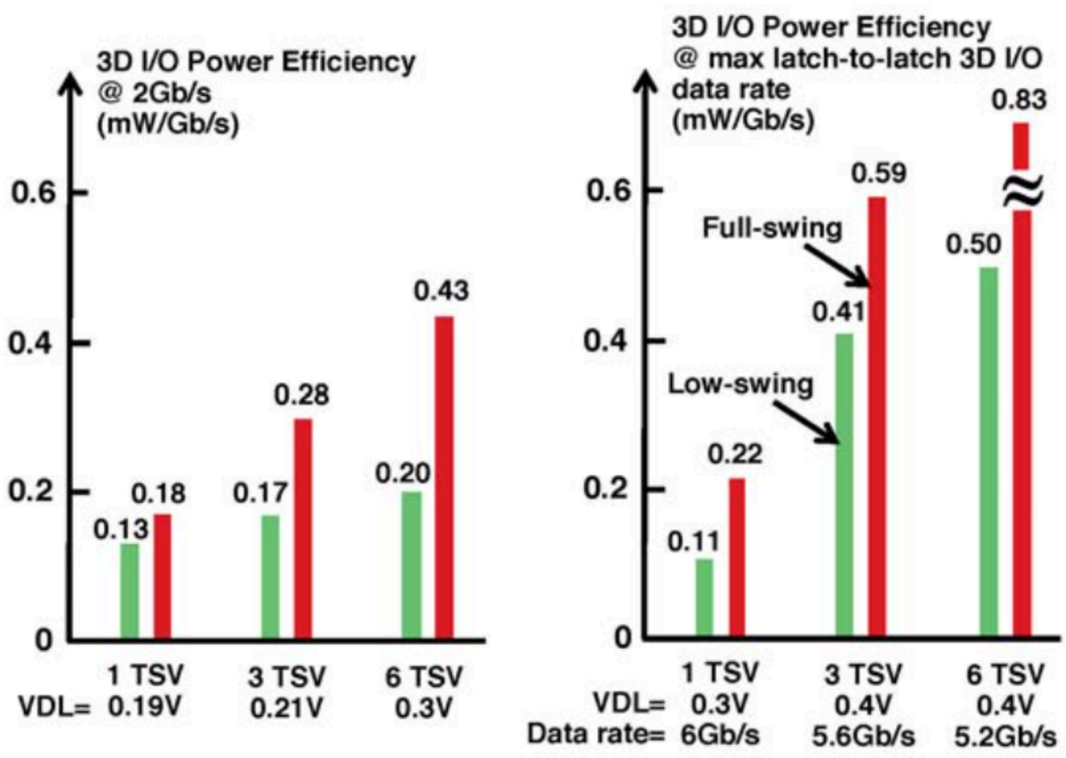


Figure 7.7.5: Measured power efficiency comparison of low-swing and full-swing latch-to-latch 3D I/O test sites.

DiRAM4 Stack

from [Pat14]