Think Different (5): Processing-In-Non-Volatile Memory

Hung-Wei Tseng

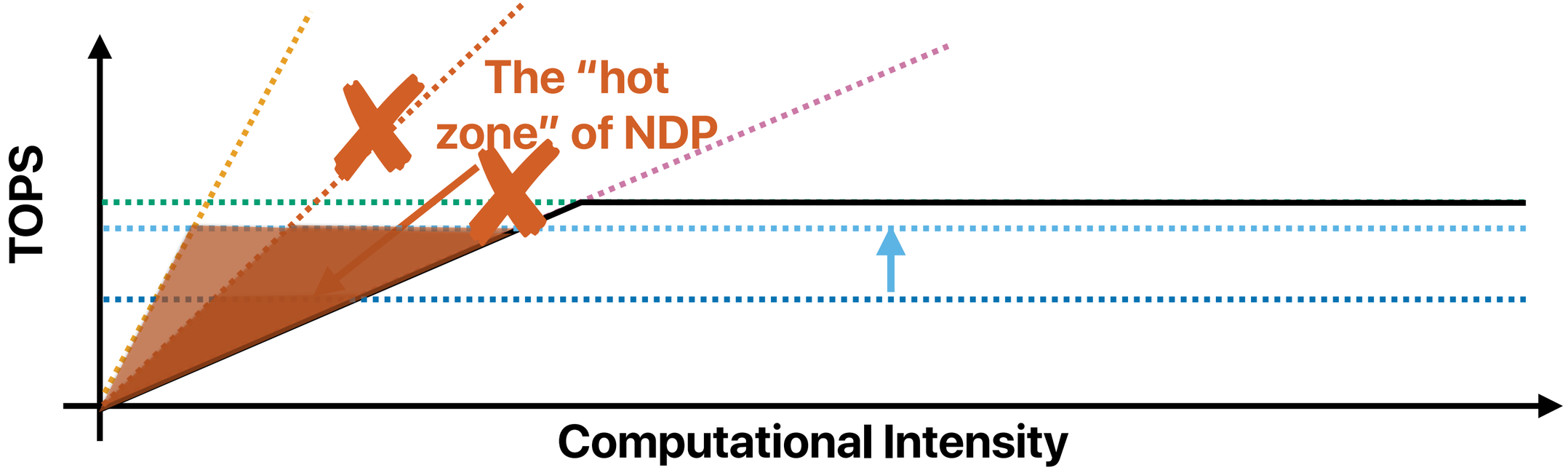
Reviewing the roofline model

~~Peak OPS of target computing resource~~

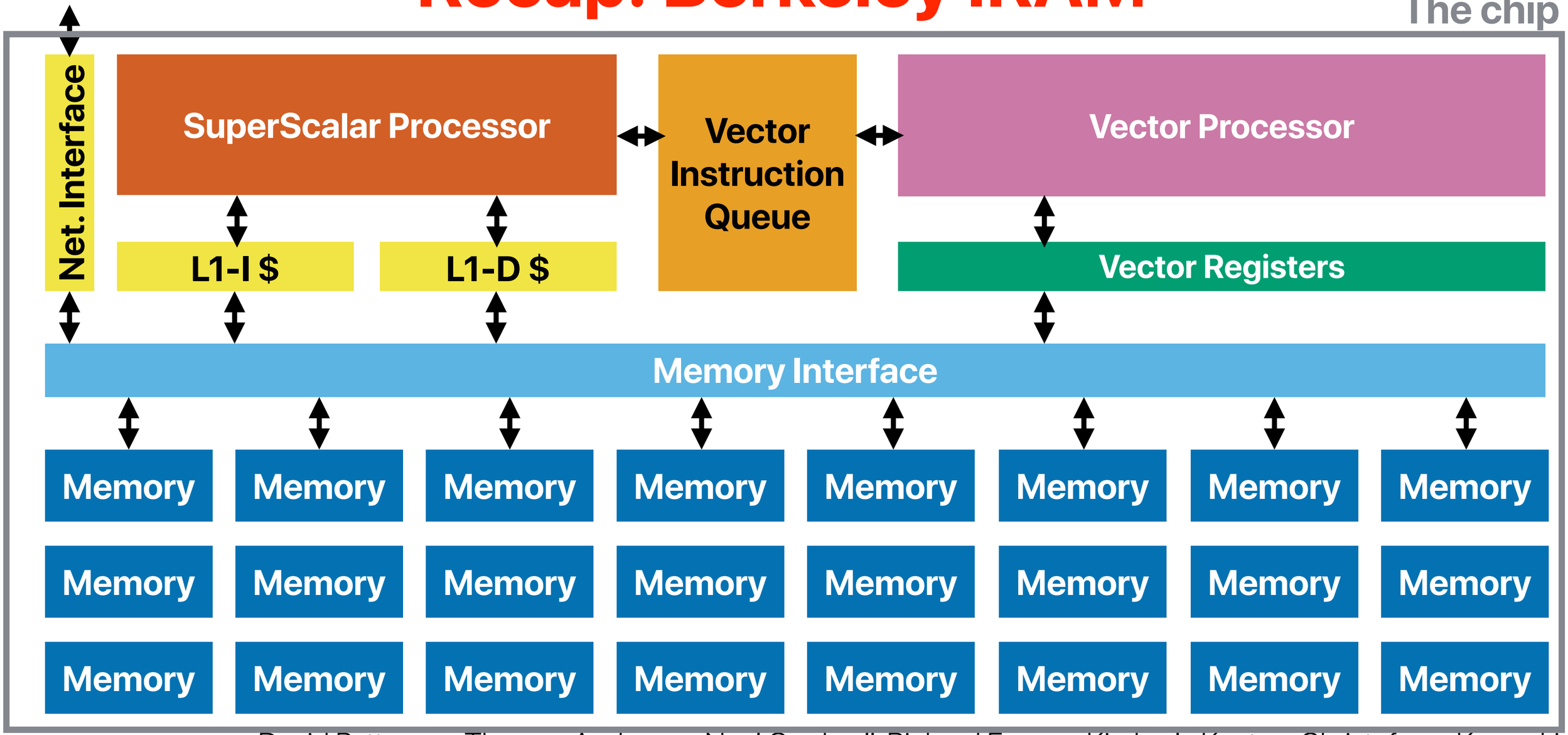
~~Peak memory bandwidth × computational intensity ÷ reduction of data volume~~

Peak OPS of target NDP device

Peak device internal bandwidth × computational intensity of NDP program



Recap: Berkeley IRAM



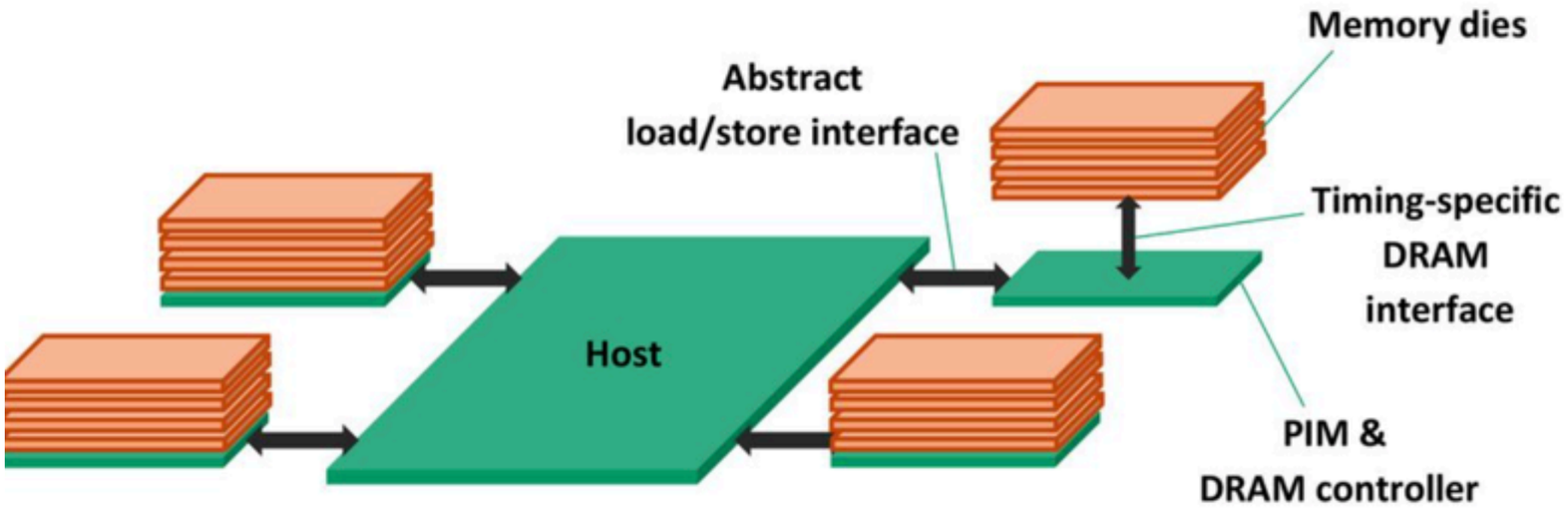
David Patterson, Thomas Anderson, Neal Cardwell, Richard Fromm, Kimberly Keeton, Christoforos Kozyrakis, Randi Thomas, and Katherine Yelick. 1997. A Case for Intelligent RAM. IEEE Micro 17, 2 (March 1997), 34–44.

Recap: PIM in the HBM era

BASELINE PIM ARCHITECTURE
AN OVERVIEW

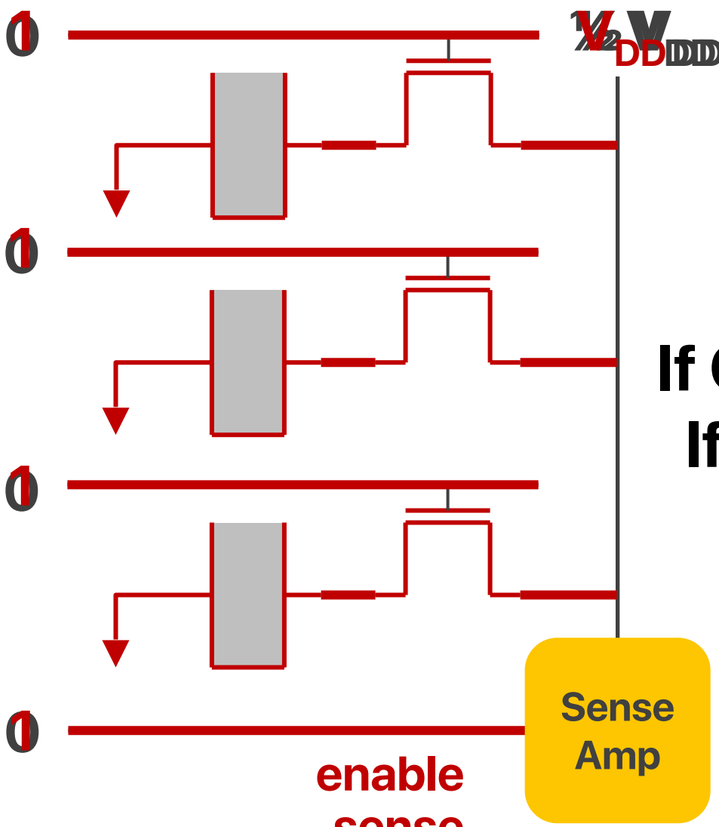
AMD

- An in-memory processor incorporated on the base die of each memory stack
- No DRAM die stacked on host processor



Dongping Zhang, Nuwan Jayasena, Alexander Lyashevsky, Joseph L. Greathouse, Lifan Xu, and Michael Ignatowski. TOP-PIM: throughput-oriented programmable processing in memory. HPDC '14. 2014

Recap: Activate Three Rows



AB + BC + AC =
C(A+B) + C′AB

If C is 0, we can compute AND
If C is 1, we can compute OR

Input			Output
A	B	C	
0	0	0	**0**
0	1	0	**0**
1	0	0	**0**
1	1	0	**1**
0	0	1	**0**
0	1	1	**1**
1	0	1	**1**
1	1	1	**1**

		A′B′	A′B	AB	AB′
	Out(A, B)	0,0	0,1	1,1	1,0
C′	0	0	0	1	0
C	1	0	1	1	1

BC AB AC

Ambit

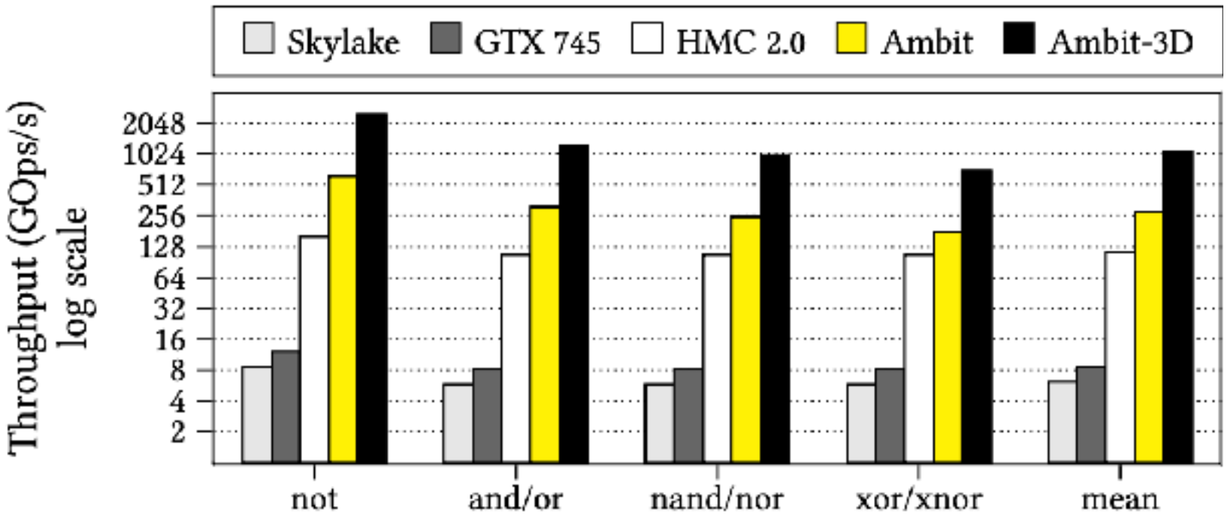


Figure 9: Throughput of bulk bitwise operations.

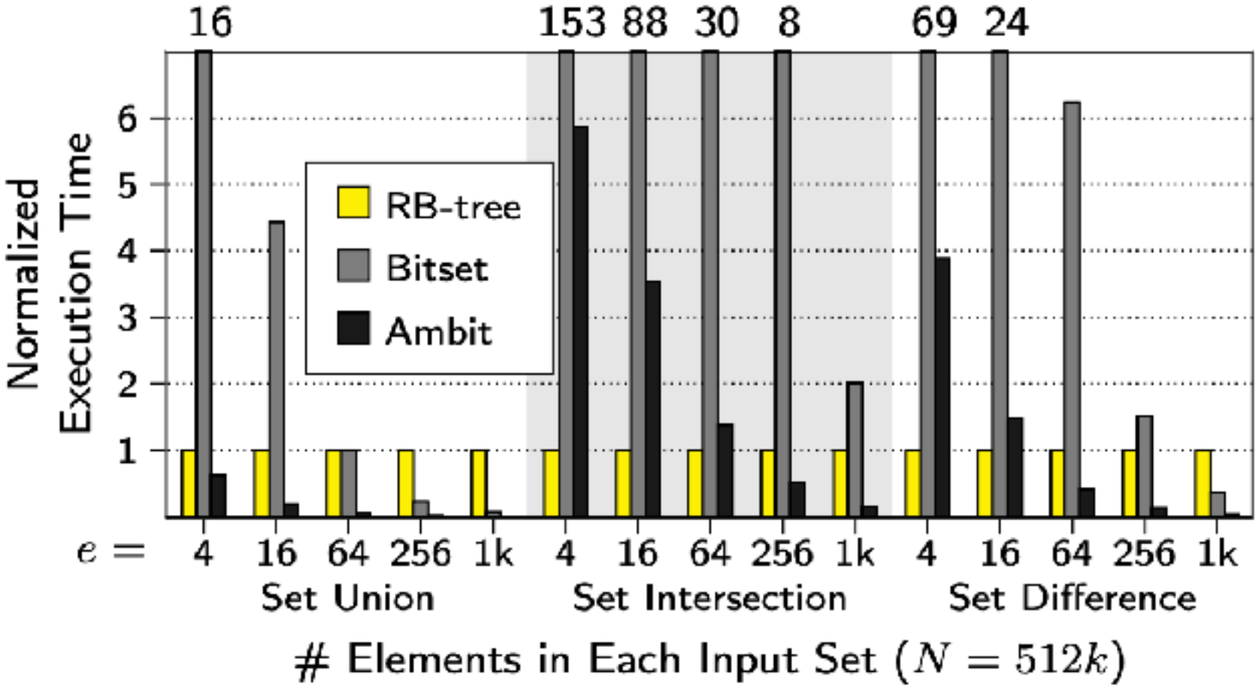


Figure 12: Performance of set operations

Vivek Seshadri, Donghyuk Lee, Thomas Mullins, Hasan Hassan, Amirali Boroumand, Jeremie Kim, Michael A. Kozuch, Onur Mutlu, Phillip B. Gibbons, and Todd C. Mowry. Ambit: in-memory accelerator for bulk bitwise operations using commodity DRAM technology. MICRO-50.

3:00

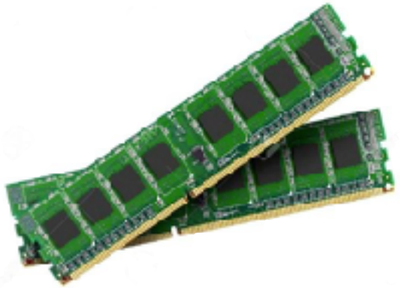
Quick exercise — can you try your best to Google and compare DRAM, Flash, Optane memory, PCM (Phase Change Memory), Resistive Random Access Memory (RRAM), Spin-Transfer Torque (STT), and Magnetoresistive RAM (MRAM)?

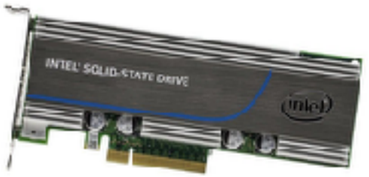
Memory technologies

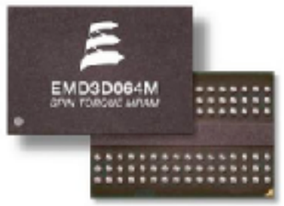
	DRAM	Flash	Optane	PCM	STT-MRAM	ReRAM
Latency						
Bandwidth						
Dollar/GB						
Endurance						
Technology						

https://www.intel.com/content/www/us/en/products/sku/201840/intel-optane-ssd-dc-p5800x-series-3-2tb-2-5in-pcie-x4-3d-xpoint/specifications.html
https://www.everspin.com/family/emd4e001g?npath=3557

Memory technologies







	DRAM	Flash	Optane	PCM	STT-MRAM	ReRAM
Latency	~ 60-100ns	~ 100 us (read) ~ 1 ms (write)	5 us (read) 6 us (write)	50 ns (read) 150 ns (write)	18 ns	10 ns
Bandwidth	~25 GB/sec per channel	3.5 GB/sec (read) 2.1 GB/sec (write)	7.2 GB/sec (read) 4.8 GB/sec (write)	?	~21 GB/sec per channel	?
Dollar/GB	3.74	0.1	3.28	?	1152 (Digikey) 144 per 1Gb	?
Endurance	10^16 cycles	< 10^6 cycles	< 10^7 cycles	~ 10^8 cycles	~ 10^12 cycles	~ 10^12 cycles
Technology	Charge	Charge	Resistive	Resistive	Resistive	Resistive

https://www.intel.com/content/www/us/en/products/sku/201840/intel-optane-ssd-dc-p5800x-series-3-2tb-2-5in-pcie-x4-3d-xpoint/specifications.html
https://www.everspin.com/family/emd4e001g?npath=3557

Charge-based v.s. resistive memory

- Charge-based memory (e.g., SRAM, DRAM, Flash)
  - Write data by capturing the charge
    - DRAM: capacitor — leakage
    - Flash: floating-gate — wear-out quickly
  - Read data by measuring the voltage level
- Resistive memory (e.g., PCM, MRAM, RRAM)
  - Write data by change the "material" properties
    - PCM: change material phase
    - STT: change magnet polarity
    - RRAM: change atomic structure/atom distance
  - Read data by measuring the resistance

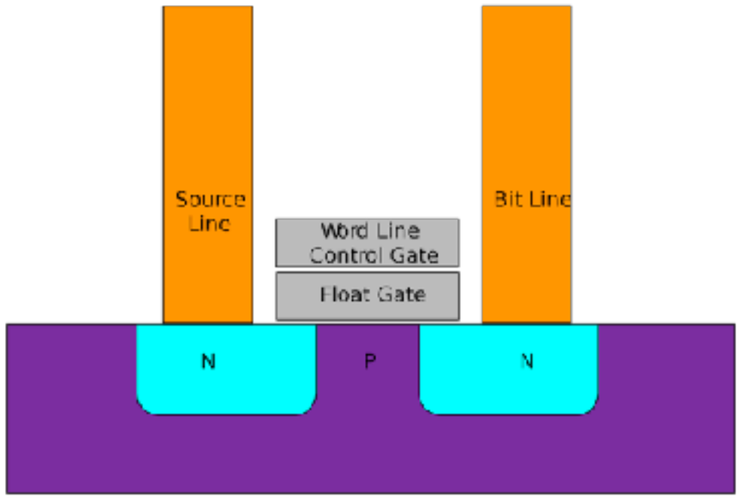
$$V_c = V_s \times e^{\frac{-t}{RC}}$$

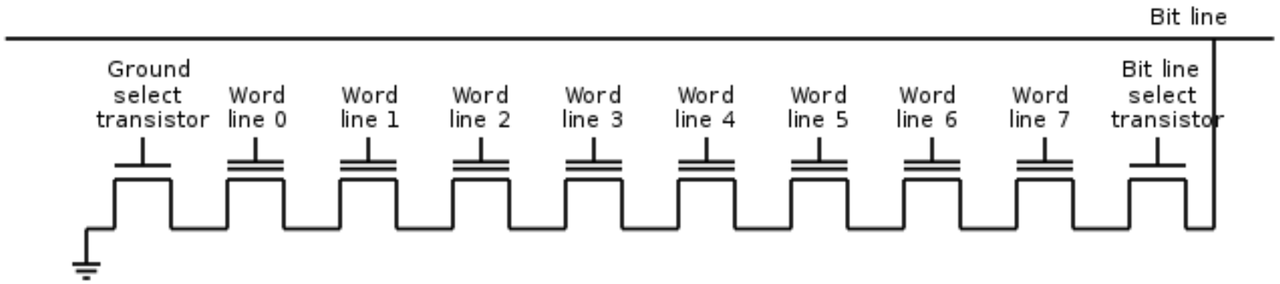
Ohm's law

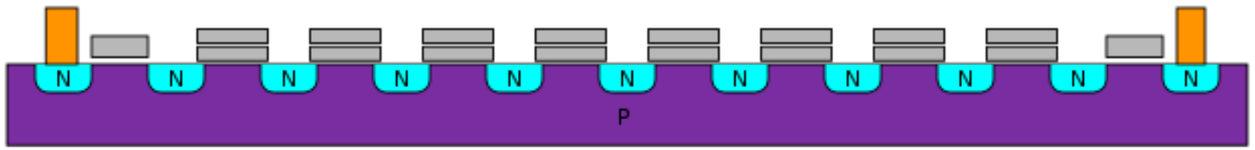
$$V = I \times R$$

Flash memory

- Floating gate made by polycrystalline silicon trap electrons
- The voltage level within the floating gate determines the value of the cell
- The floating gates will wear out eventually

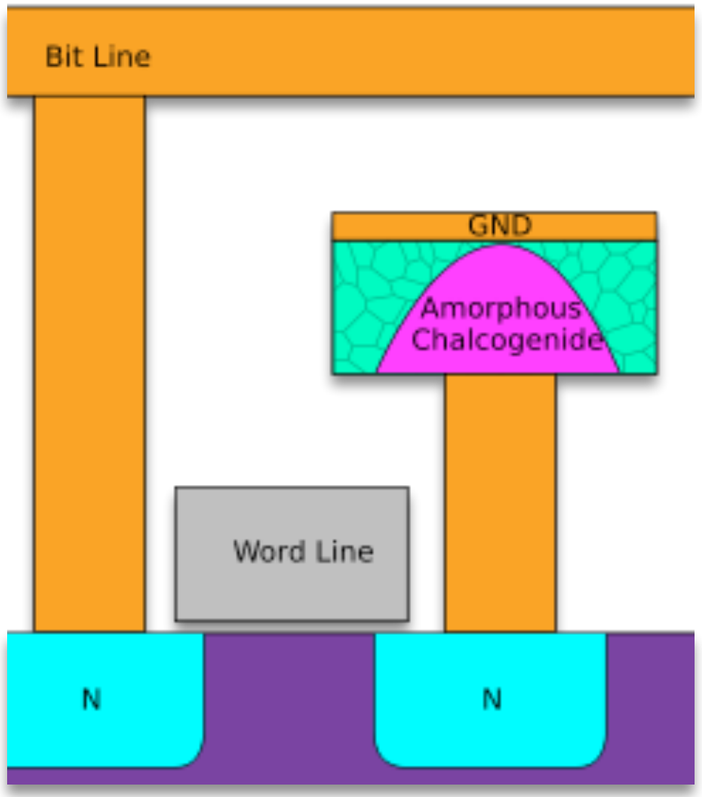






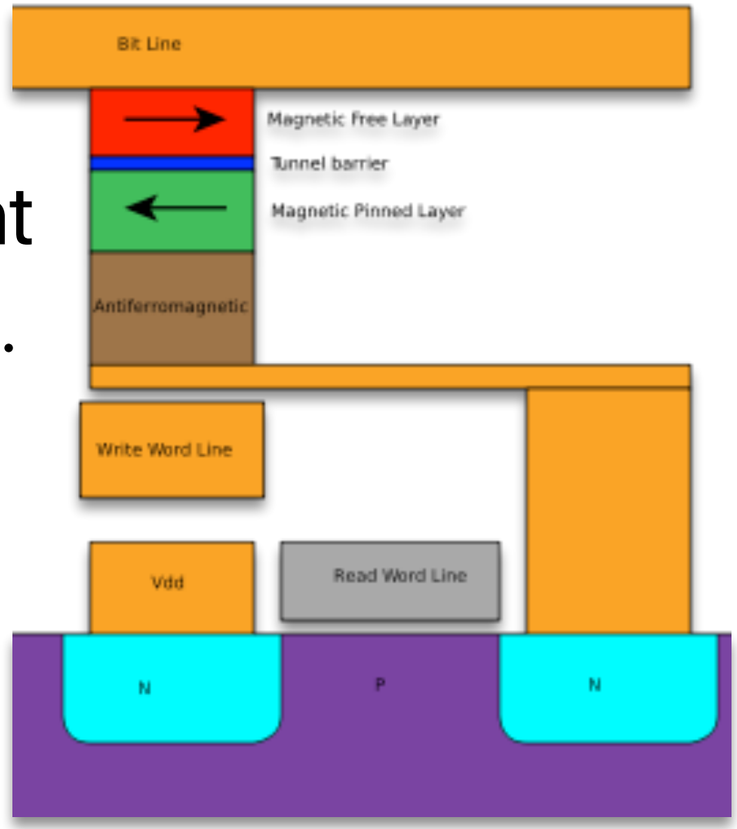
Phase change memory

- The bit is stored in the crystal structure of a tiny spec of metal.
- To write, it melts the metal (650C)
  - let it cool quickly or slowly to set the value
  - Crystaline and amorphous states have different resistance
    - Amorphous: Low optical reflexivity and high electrical resistivity
    - Crystalline: High optical reflexivity and low electrical resistivity



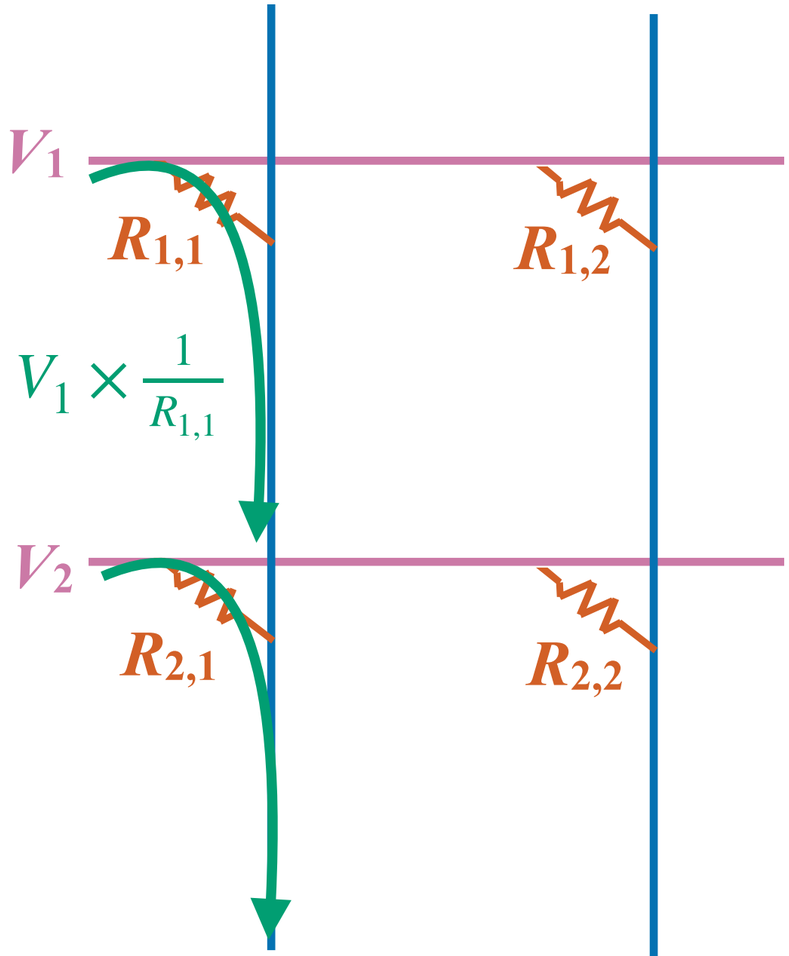
Spin-torque transfer

- Bits stored as magnetic orientation of a thin film
- Change the state using polarized electrons (!)
- Depending on polarization, resistance differs
- More complex cell structure
- Great promise — potential DRAM replacement
  - Roughly the same speed, power, and bandwidth.
  - But it's durable!



“In”-NVM processing

ReRAM



$$I_{1,1} = V_1 \times \frac{1}{R_{1,1}}$$

$$I_1 = V_1 \times \frac{1}{R_{1,1}} + V_2 \times \frac{1}{R_{2,1}} \qquad I_2 = V_1 \times \frac{1}{R_{2,1}} + V_2 \times \frac{1}{R_{2,2}}$$

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} \times \begin{bmatrix} R_{1,1}^{-1} & R_{1,2}^{-1} & R_{1,3}^{-1} & R_{1,4}^{-1} \\ R_{2,1}^{-1} & R_{2,2}^{-1} & R_{2,3}^{-1} & R_{2,4}^{-1} \\ R_{3,1}^{-1} & R_{3,2}^{-1} & R_{3,3}^{-1} & R_{3,4}^{-1} \\ R_{4,1}^{-1} & R_{4,2}^{-1} & R_{4,3}^{-1} & R_{4,4}^{-1} \end{bmatrix}$$

3:00

What's the limitation of ReRAM-based matrix multiplier? What applications can tolerate these limitations?

Limitations of ReRAM-based accelerator

Limitations of ReRAM-based accelerator

- Limited Precision
- A/D and D/A Conversion
  - Area and power increases exponentially with ADC resolution and frequency
  - Large area, Power hungry e.g., 98% of the total area and 89% of the total power
- Array Size and Routing
  - Wire dominates energy for array size of 1k × 1k
  - IR drop along wire can degrade read accuracy
- Write/programming energy
  - Multiple pulses can be costly
- Variations & Yield
  - Device-to-device, cycle-to-cycle
  - Non-linear conductance across range

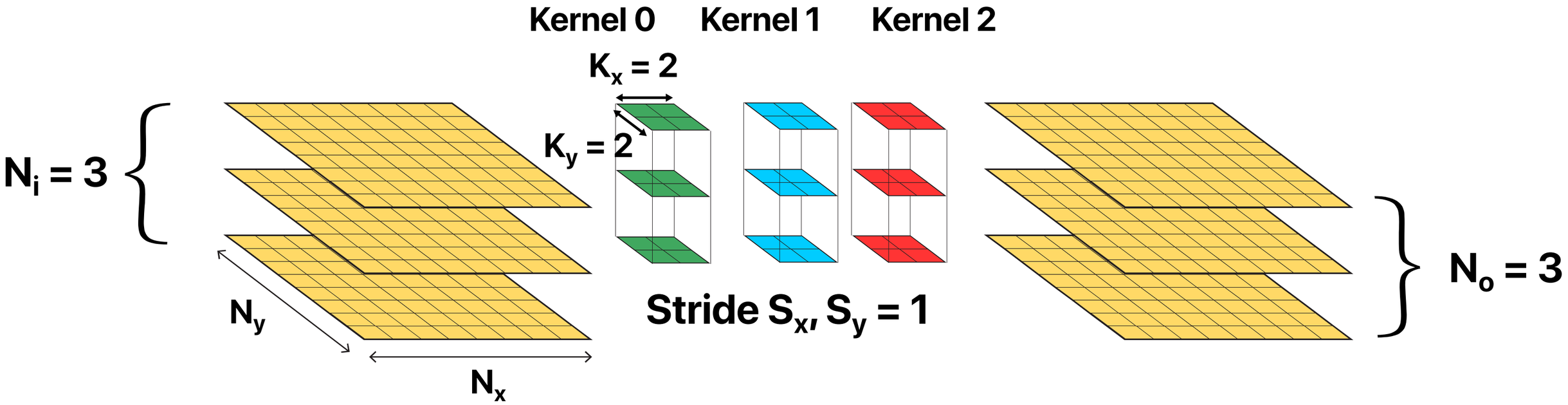
References

- ADC/DAC optimizations
  - H. Yun, H. Shin, M. Kang and L. -S. Kim, "Optimizing ADC Utilization through Value-Aware Bypass in ReRAM-based DNN Accelerator," 2021 58th ACM/IEEE Design Automation Conference (DAC), 2021, pp. 1087-1092, doi: 10.1109/DAC18074.2021.9586140.
  - Qilin Zheng, Zongwei Wang, Zishun Feng, Bonan Yan, Yimao Cai, Ru Huang, Yiran Chen, Chia-Lin Yang, and Hai (Helen) Li. 2020. Lattice: an ADC/DAC-less ReRAM-based processing-in-memory architecture for accelerating deep convolution neural networks. In Proceedings of the 57th ACM/EDAC/IEEE Design Automation Conference (DAC '20). IEEE Press, Article 190, 1–6.
- Digital PIM
  - Mohsen Imani, Saransh Gupta, Yeseong Kim, and Tajana Rosing. 2019. FloatPIM: in-memory acceleration of deep neural network training with high precision. In Proceedings of the 46th International Symposium on Computer Architecture (ISCA '19). Association for Computing Machinery, New York, NY, USA, 802–815. https://doi.org/10.1145/3307650.3322237

Applications of ReRAM-based accelerator

- NN
  - Ali Shafiee, Anirban Nag, Naveen Muralimanohar, Rajeev Balasubramonian, John Paul Strachan, Miao Hu, R. Stanley Williams, and Vivek Srikumar. 2016. ISAAC: a convolutional neural network accelerator with in-situ analog arithmetic in crossbars. In ISCA '16. 2016
  - Ping Chi, Shuangchen Li, Cong Xu, Tao Zhang, Jishen Zhao, Yongpan Liu, Yu Wang, and Yuan Xie. PRIME: a novel processing-in-memory architecture for neural network computation in ReRAM-based main memory. In ISCA '16. 2016
  - Song, Linghao, Xuehai Qian, Hai Li, and Yiran Chen. Pipelayer: A pipelined reram-based accelerator for deep learning. In HPCA 2017.
  - Mohsen Imani, Saransh Gupta, Yeseong Kim, and Tajana Rosing. 2019. FloatPIM: in-memory acceleration of deep neural network training with high precision. ISCA. 2019.

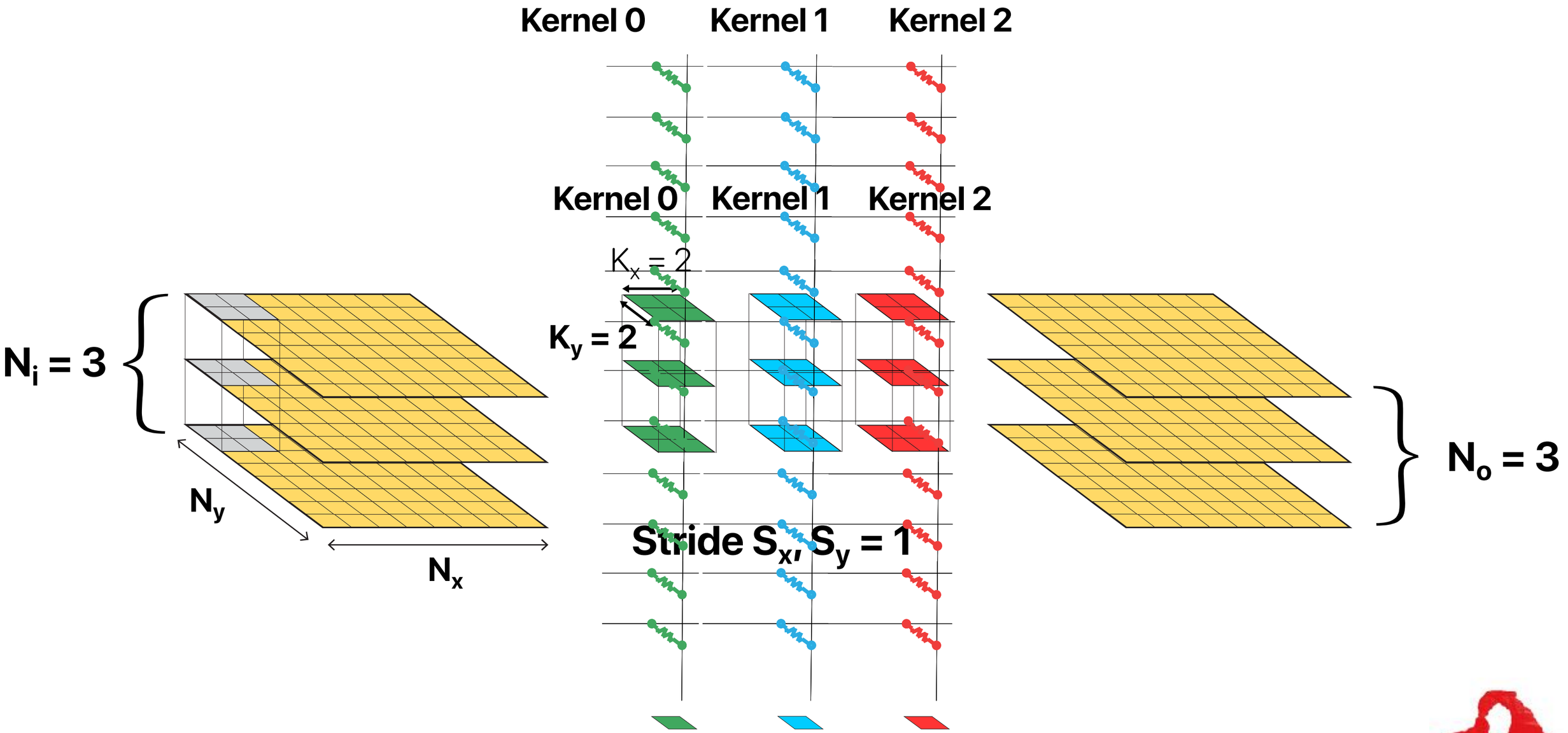
Convolution



ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars, ISCA 2016

UTAH ARCH

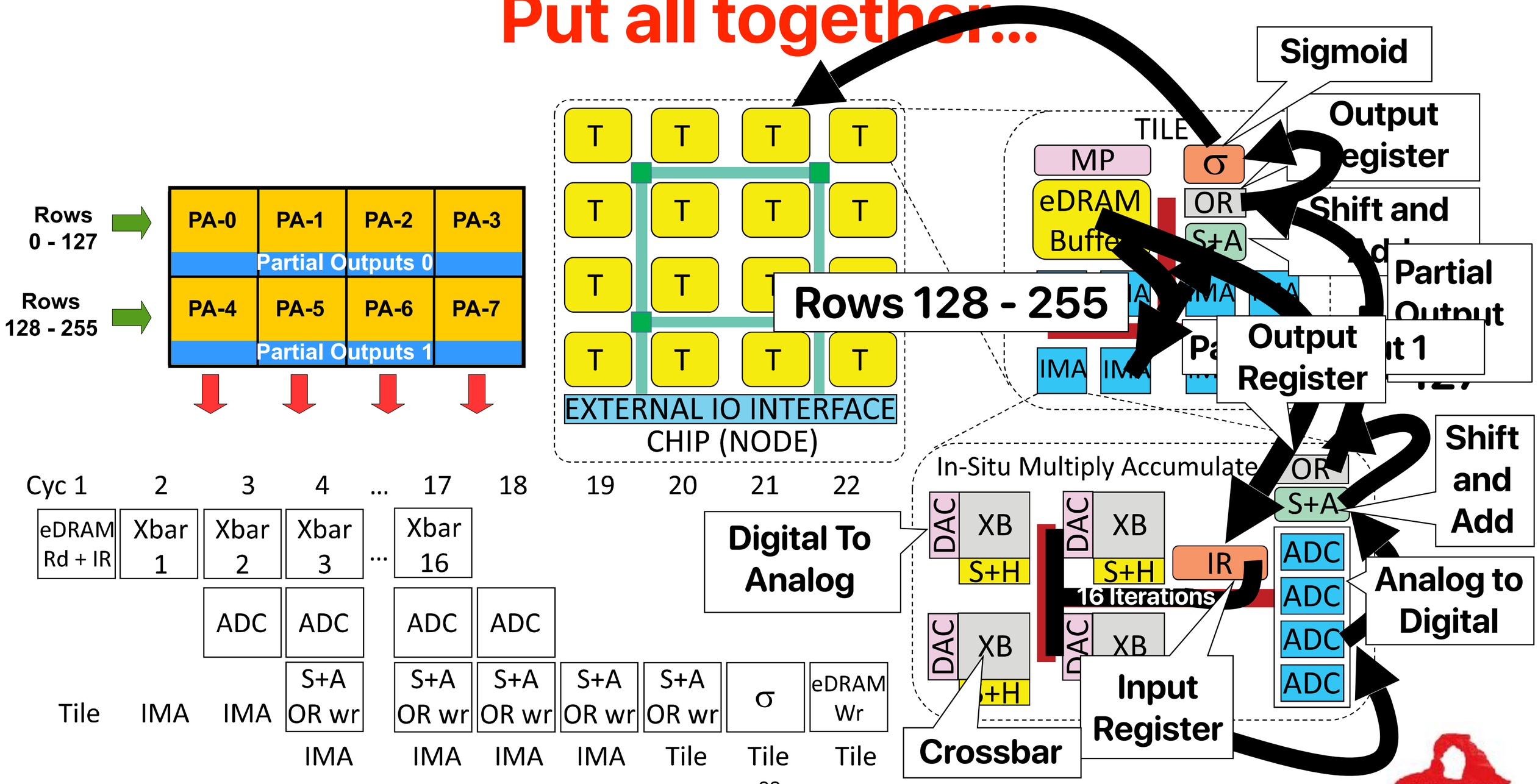
Convolution in ReRAM



ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars, ISCA 2016

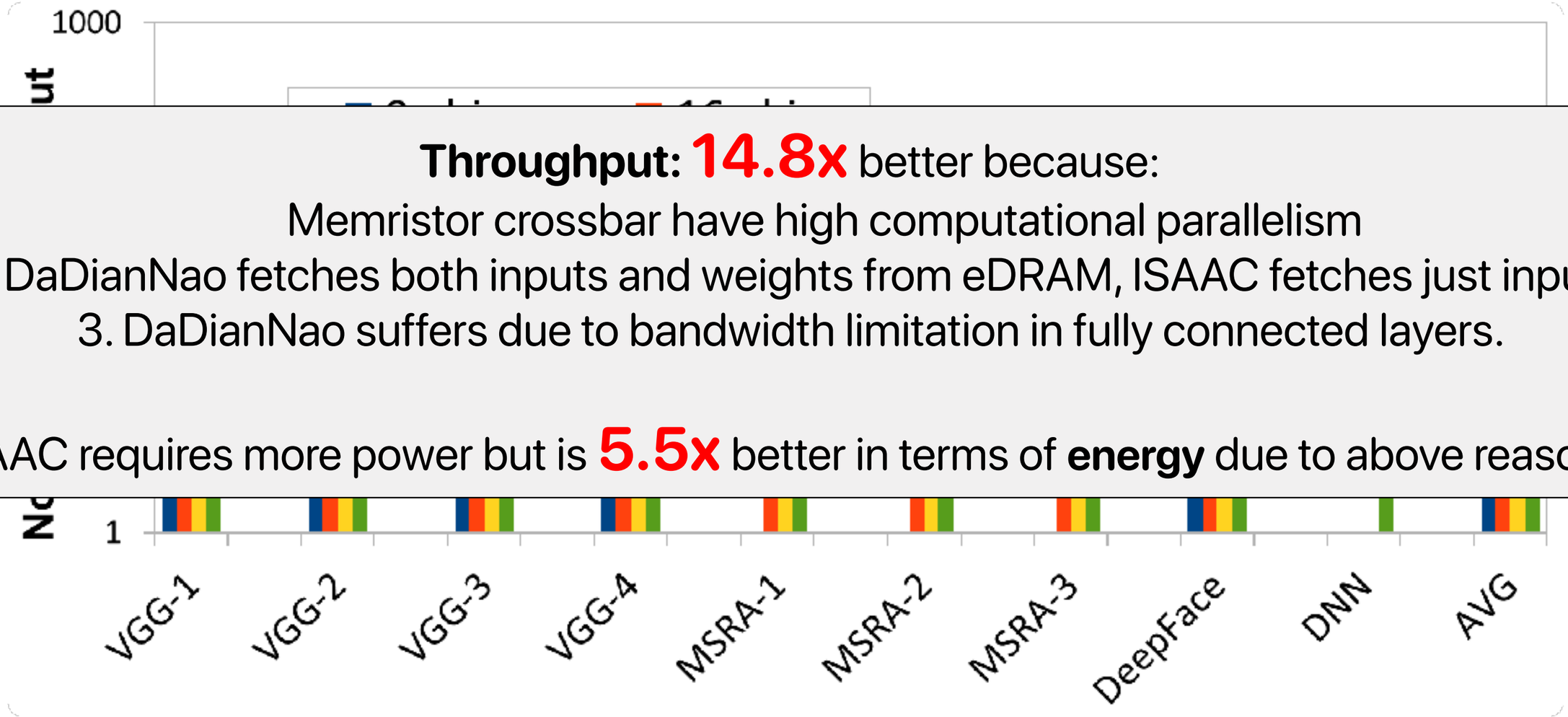
UTAH ARCH

Put all together…



ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars, ISCA 2016

UTAH ARCH



Throughput: **14.8x** better because:

1. Memristor crossbar have high computational parallelism
2. DaDianNao fetches both inputs and weights from eDRAM, ISAAC fetches just inputs
3. DaDianNao suffers due to bandwidth limitation in fully connected layers.

ISAAC requires more power but is **5.5x** better in terms of **energy** due to above reasons.

Deep Neural Net Benchmarks

ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars, ISCA 2016

UTAH ARCH

TABLE I

COMPARISON TABLE OF RECENT ANALOG IMC-BASED MVM/MULTIPLY-ACCUMULATE (MAC)-OPERATION ACCELERATORS

Metric	This work	ISSCC'21 [45]	ISSCC'21 [44]	ISSCC'20 [46]
CMOS technology	14 nm	22 nm	16 nm	7 nm
Memory technology	PCM	ReRAM	SRAM	SRAM
Non-volatile	Yes	Yes	No	No
Operating Voltage in V	0.8	0.8	0.8	0.8
Operation Frequency	1 GHz	-	200 MHz	-
ADC design	CCO-based ADC	Sense amplifier	8bit SAR ADC	4bit Flash-ADC
Memory size	65.5 K	4 M	4.5 MB	4 KB
Unit-cell	8T4R	1T1R	10T1C	8T
Number of input/weight/output-bits	8b/Analog/8b	8b/8b/14b	4b/4b/8b	4b/4b/4b
Peak Throughput (TOPS)	1.008	0.035	11.8 | 5.90[1]	0.372 | 0.186[1]
Energy Efficiency (TOPS/W)	10.5	11.91	121 | 60.5[1]	351 | 175.5[1]
Area Efficiency (TOPS/mm^2)	1.59	0.013	2.67 | 1.34[1]	116.3 | 58.13[1]

and functionality of the digital ADC calibration procedure is described in detail and the MVM accuracy is quantified. Finally, the measured classification accuracies of deep learning (DL) inference applications on the MNIST and CIFAR-10 datasets, when two IMC cores are employed, are presented. For a performance density of 1.59 TOPS/mm^2, a measured energy efficiency of 10.5 TOPS/W, at a main clock frequency of 1 GHz, is achieved.

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9696185

3:00

If you wants to support 50Gbps network, how fast does your processor needs to be to saturate the network?

Network protocol stack

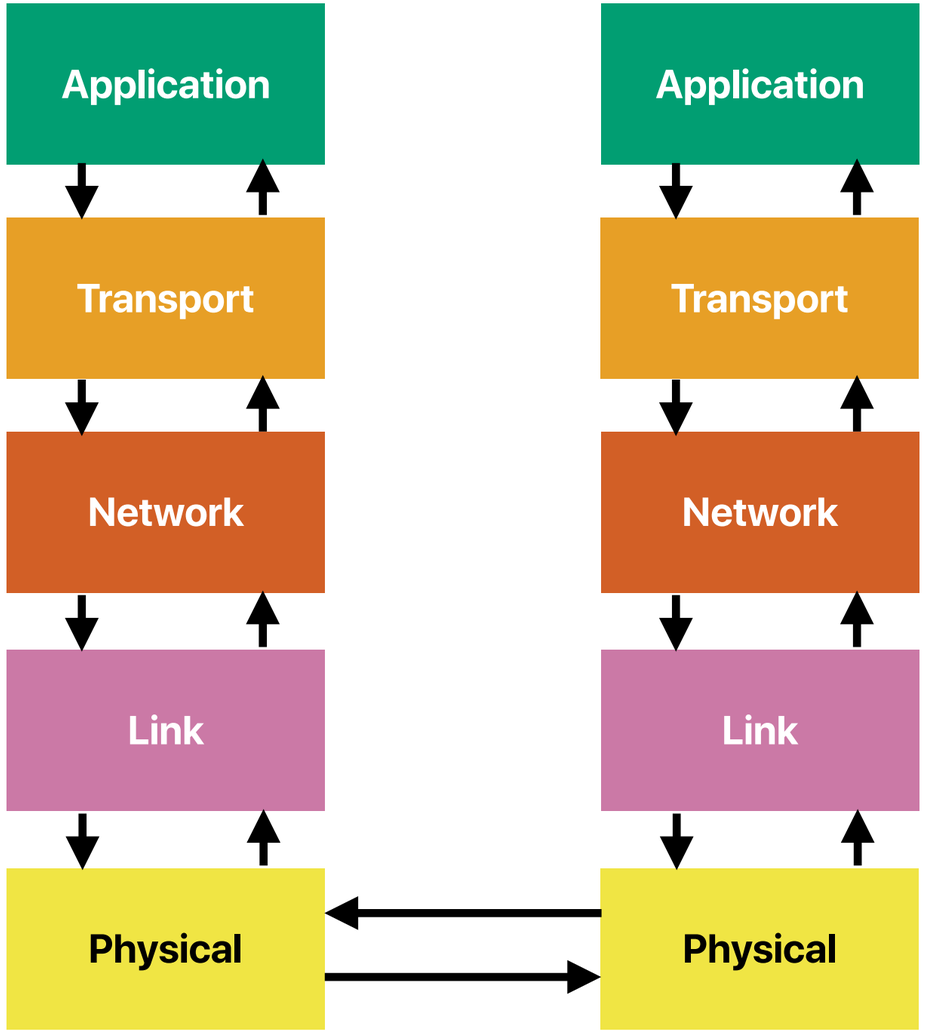
How do applications(e.g., server/client) interpret data: HTTPS, FTP, SSH, RTSP …

How do applications connect to each other (end-to-end reliability): TCP, UDP, RTP

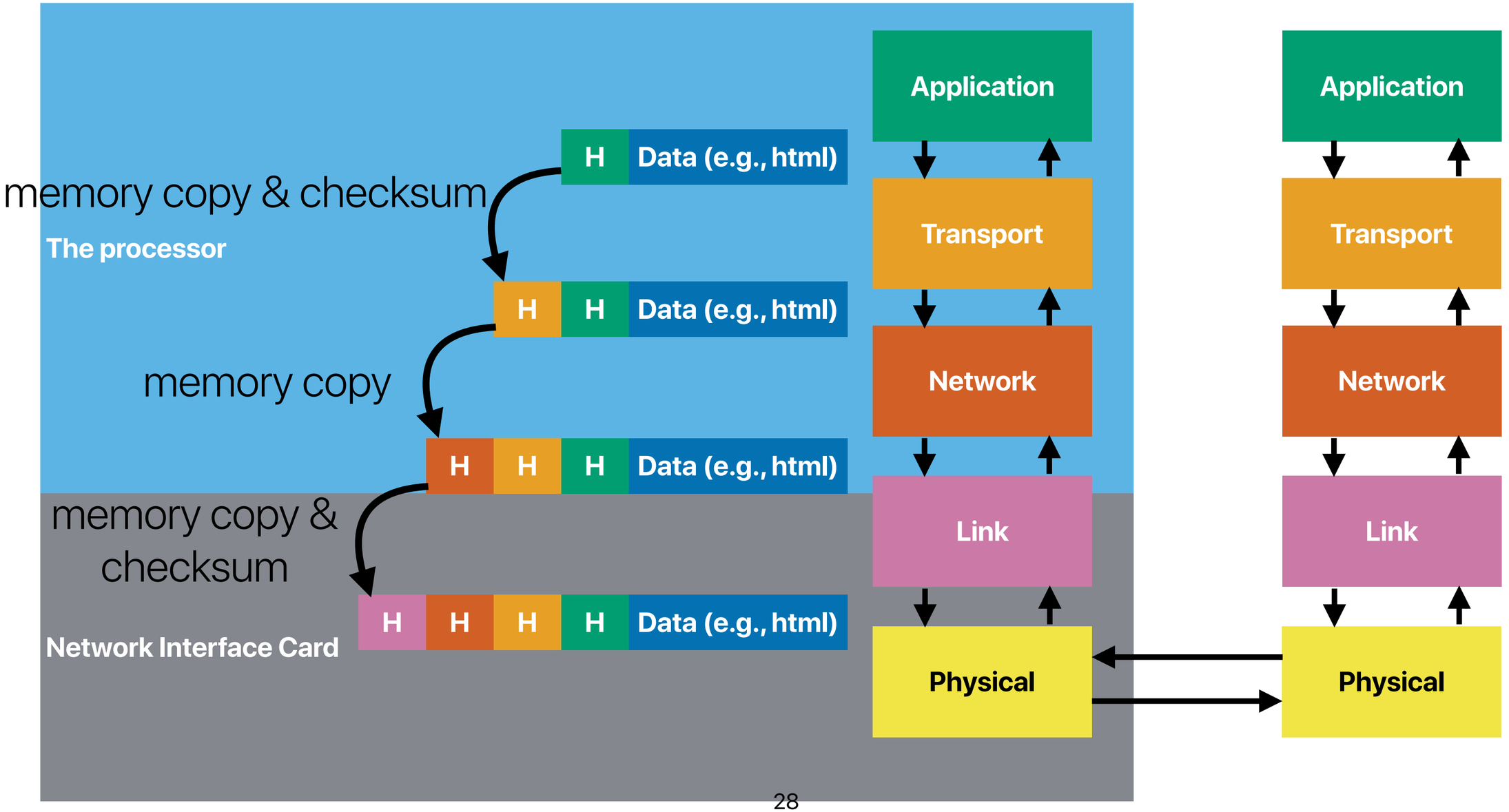
How do computers find each other: IP, ARP, ICMP

How do a computer use the media:
IEEE 802.3 (ethernet), IEEE 802.11 (WiFi)

How do the computer encode data on the media



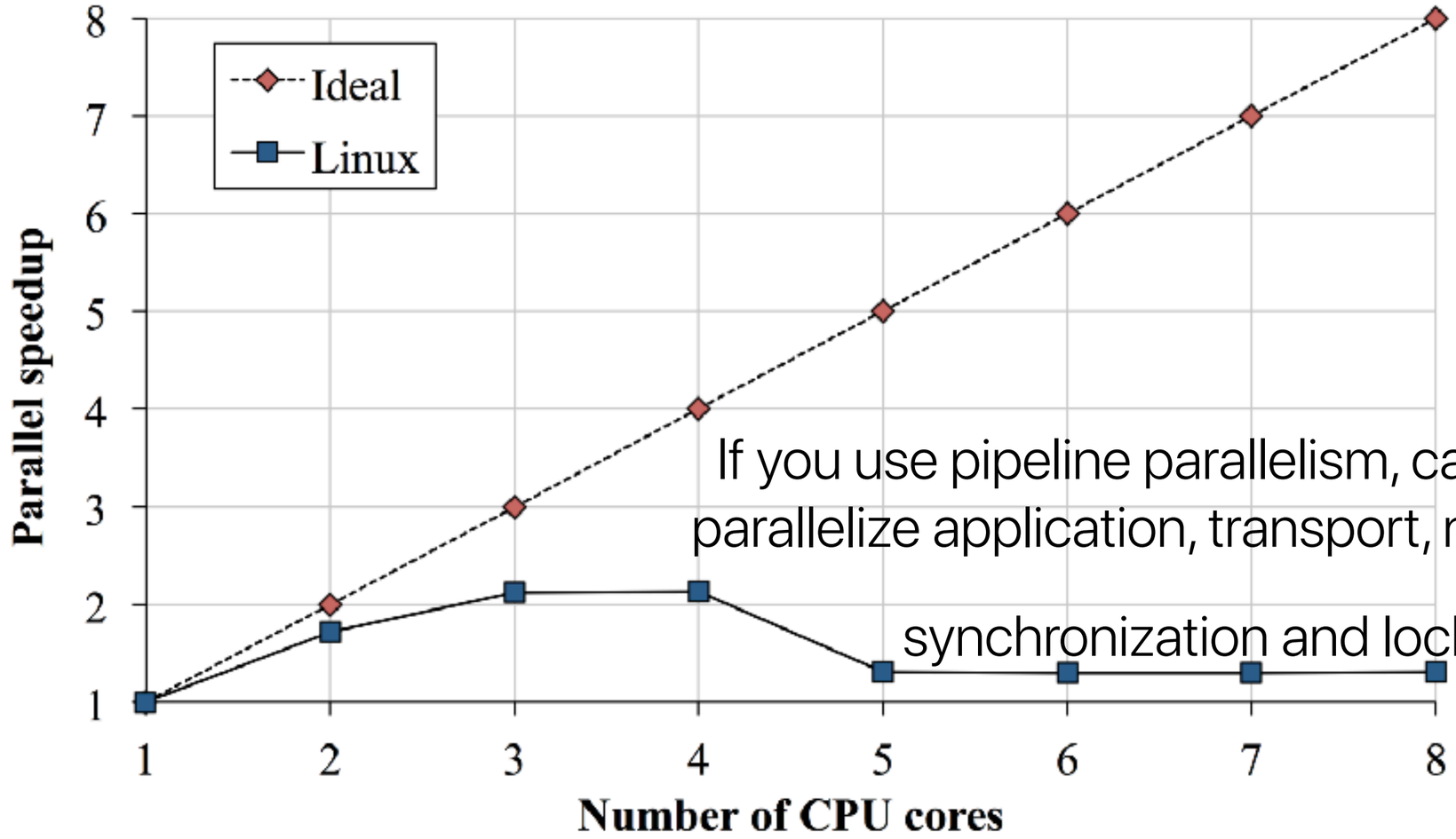
Network protocol stack



What do you need a processor for networks?

- Protocol processing
  - Checksum — memory accesses, computation
  - Encryption/decryption — memory accesses, computation
- 50 Gbps == 6.25GB/sec "goodput"
  - Multiple memory copies — consumes bandwidth, using load/store instructions
  - DDR4 DRAM bandwidth == 25GB/sec per channel

How does packet processing look like in multicore processors?



If you use pipeline parallelism, can only parallelize application, transport, network

synchronization and locking overhead

Figure 2.2: The parallel speedup of the Linux network stack, measured with a RPC-like workload on a 8-core server. For the experiment we generated TCP client connections, each of which exchanges a pair of 64-byte dummy request and request.

https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-112.pdf

Think different

DDR4
26 GB/sec

CPU-Memory BUS

Memory Controller

Processor

DRAM

We have processors near data

1 GB/sec

CPU

PCIe Root Complex

x4
4GB/sec

Processor

SSD

22 GB/sec

GPU

105 GB/sec

Processor

NIC

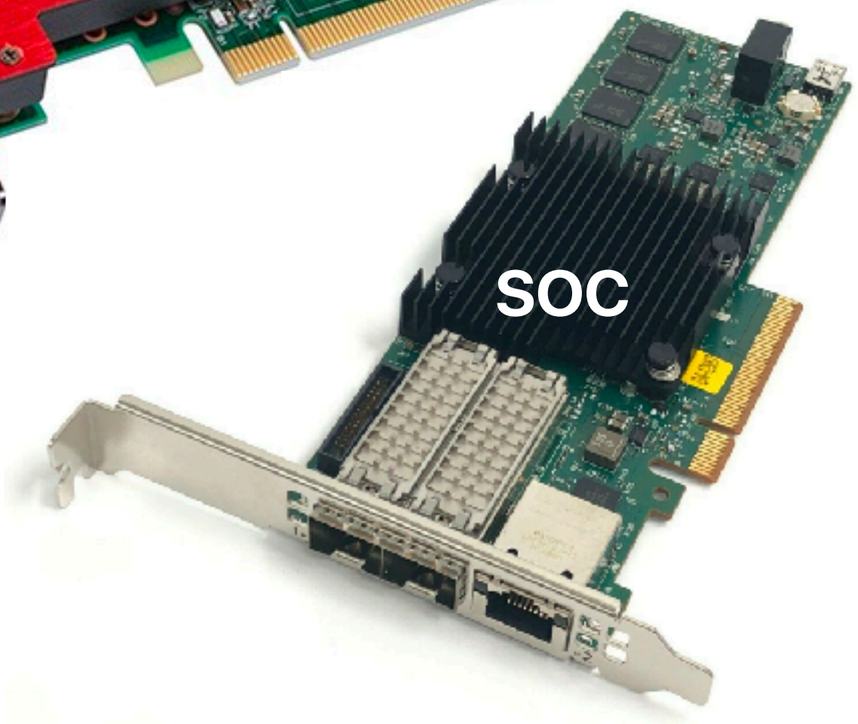
TPU

FPGA

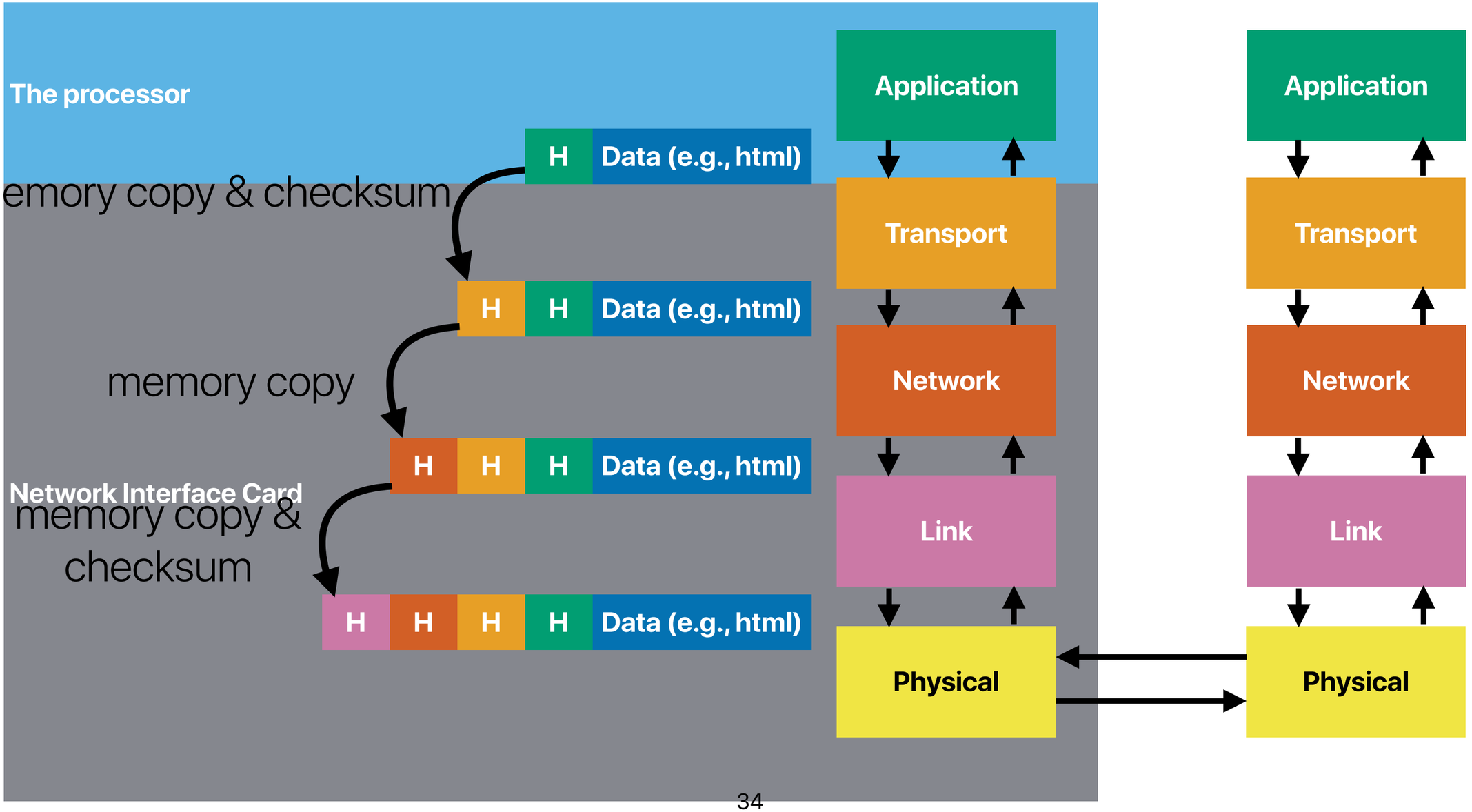
XPU

In-network processing

SmartNICs

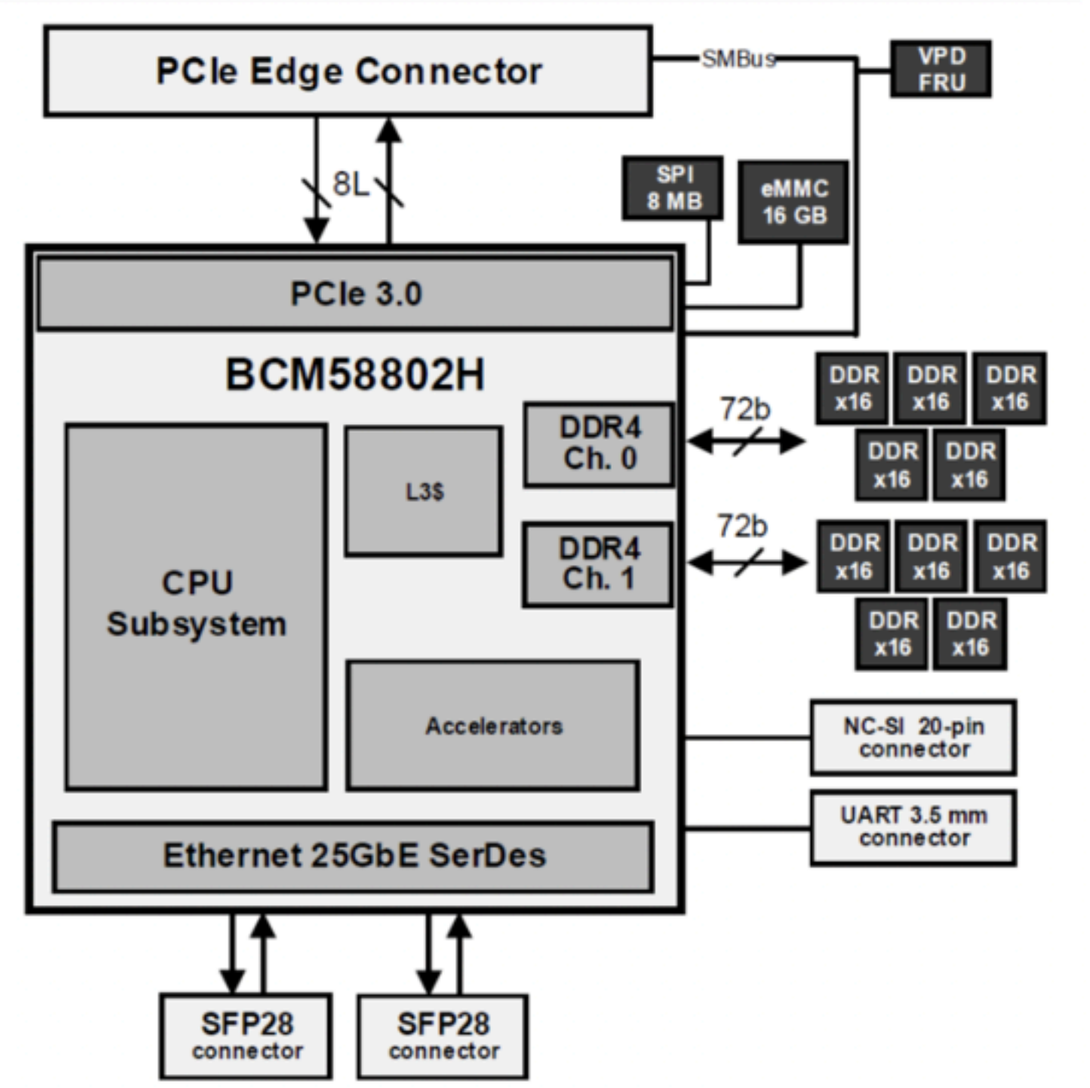


Network protocol stack w/ SmartNICs



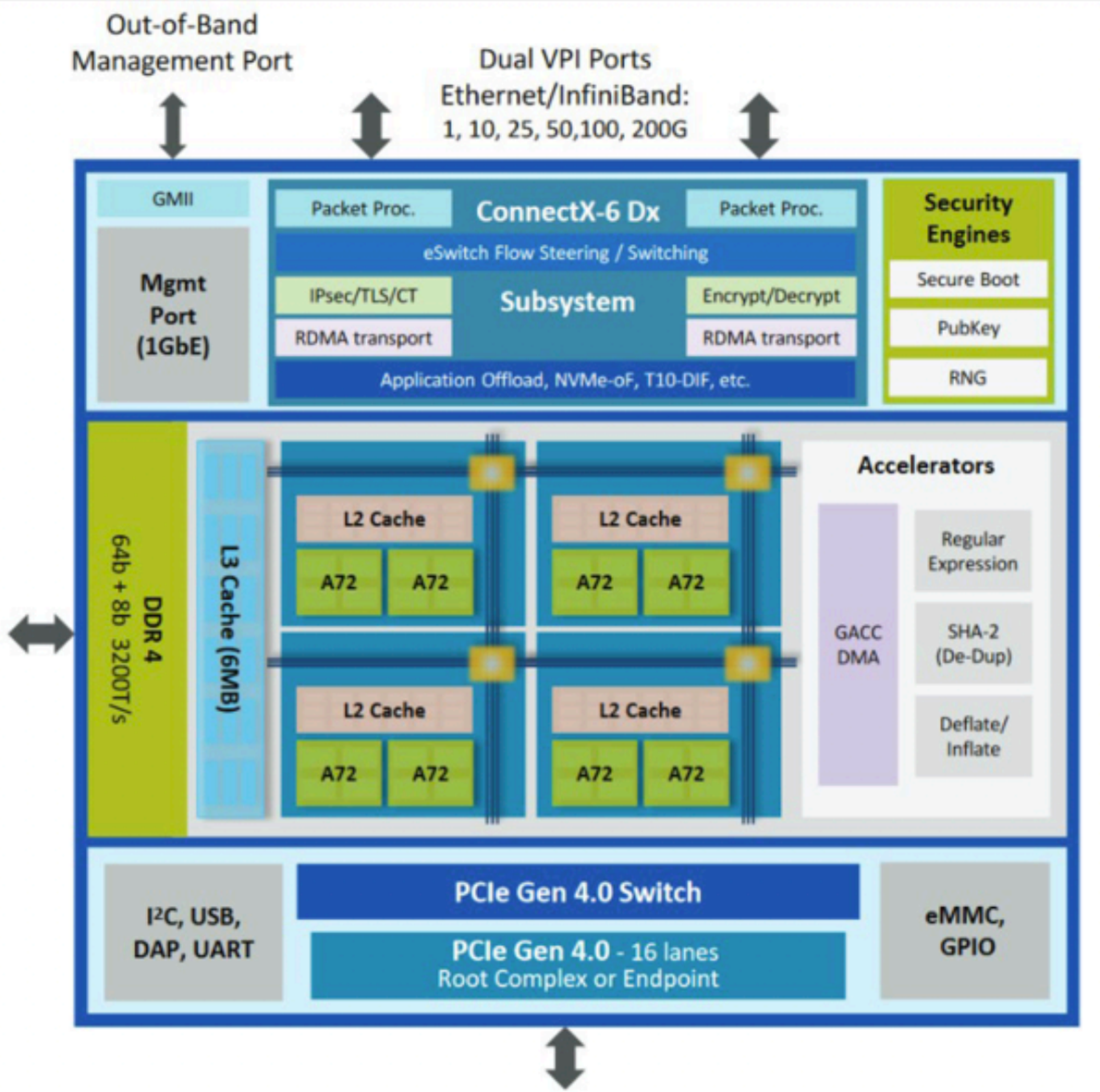
SmartNICs Top Six: Stingray/Broadcom

- ASIC Based
  - Flow classifier
  - 8 ARM Cores
  - IP Accelerators
- Memory
  - 2 Banks DDR4



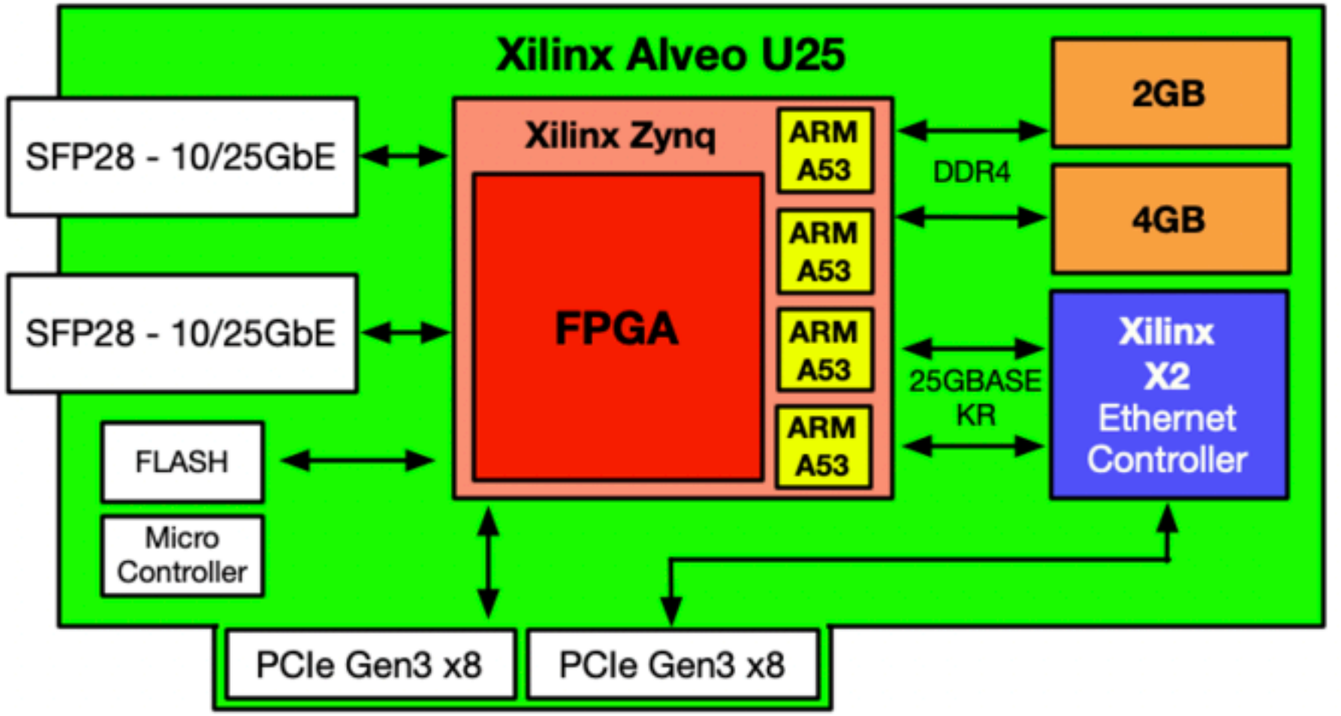
SmartNICs Top Six: BlueField2/NVIDIA

- ASIC Based
  - ConnectX-6 Dx
  - 8 ARM Cores
  - IP Accelerators
- Memory
  - DDR4

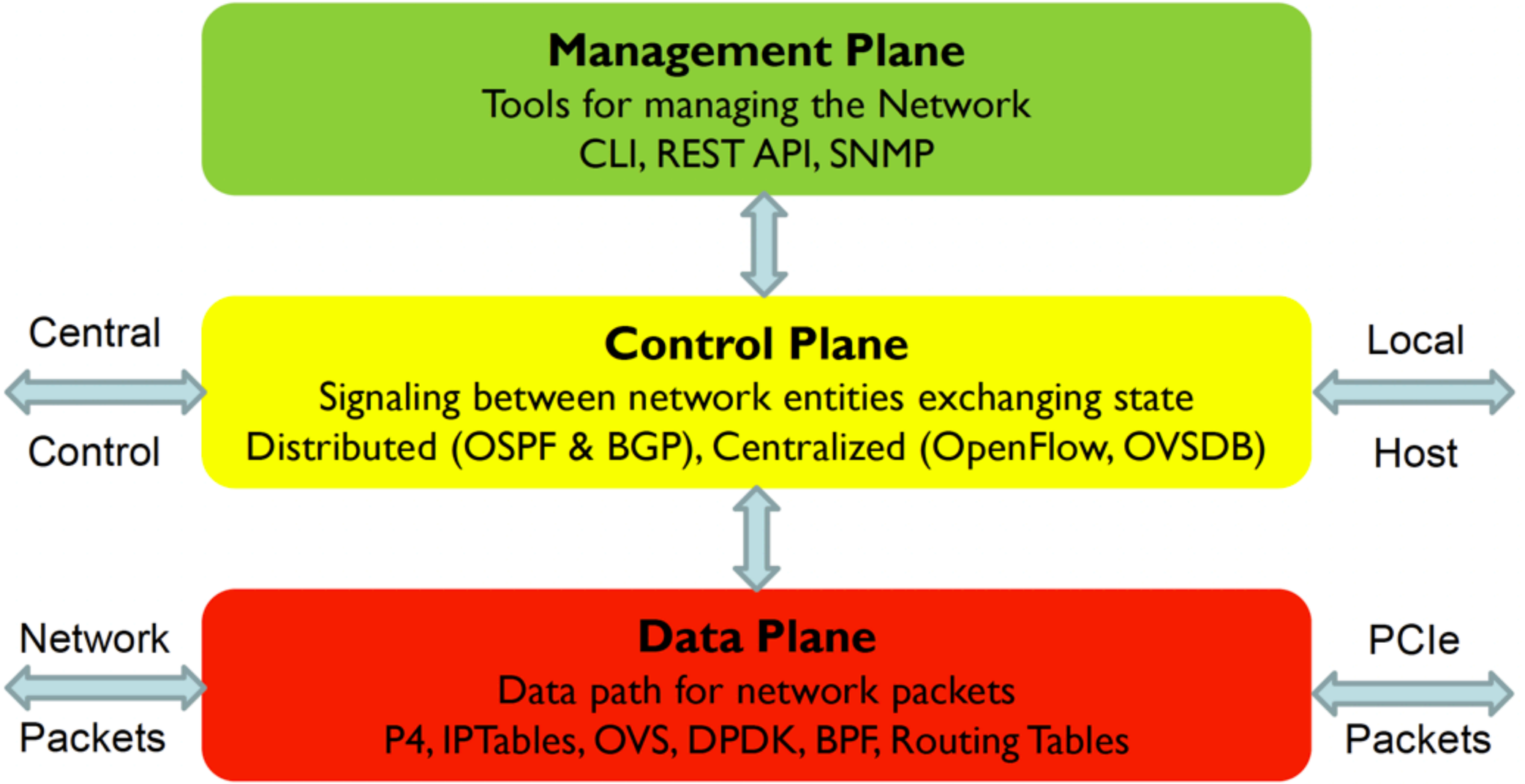


SmartNICs Top Six: U25/Xilinx

- ASIC – X2
- FPGA – Zynq
  - Large FPGA
  - 4 ARM Cores
- Memory
  - 2x DDR4



SmartNICs: Architecture



*Note Borrowed from: Docker Networking: Control Plane Data Plane Presentation

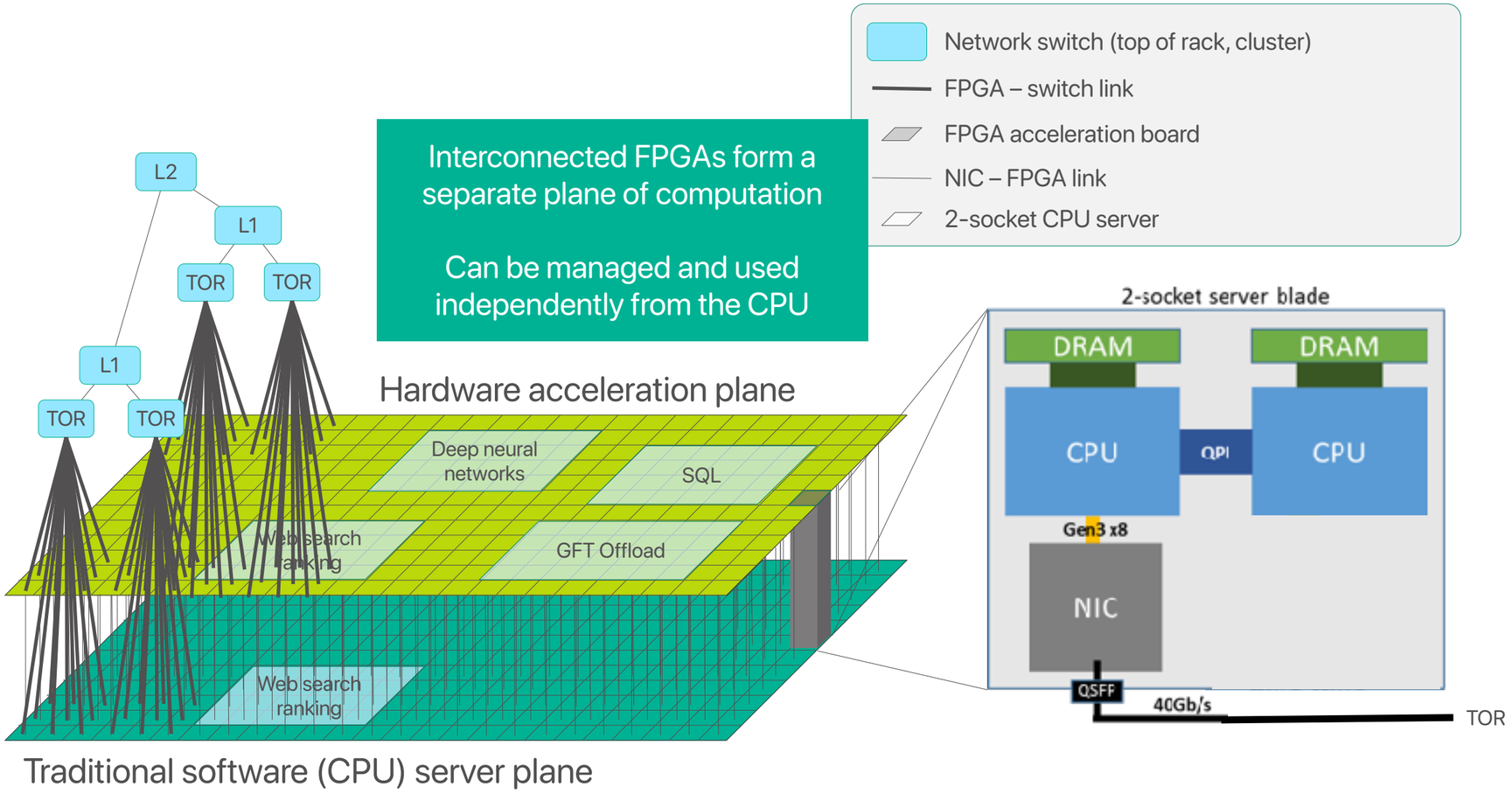
SmartNICs: Building Blocks

- Hardware
  - Cores, IP, Programable Logic, Memory & Interconnect
- Protocols
  - PCIe, CXL, CCIX, Ethernet, IB, UDP, TCP, HTML/3 & QUIC
- Ecosystem
  - Languages, SDK & App Stores

A Cloud-Scale Acceleration Architecture

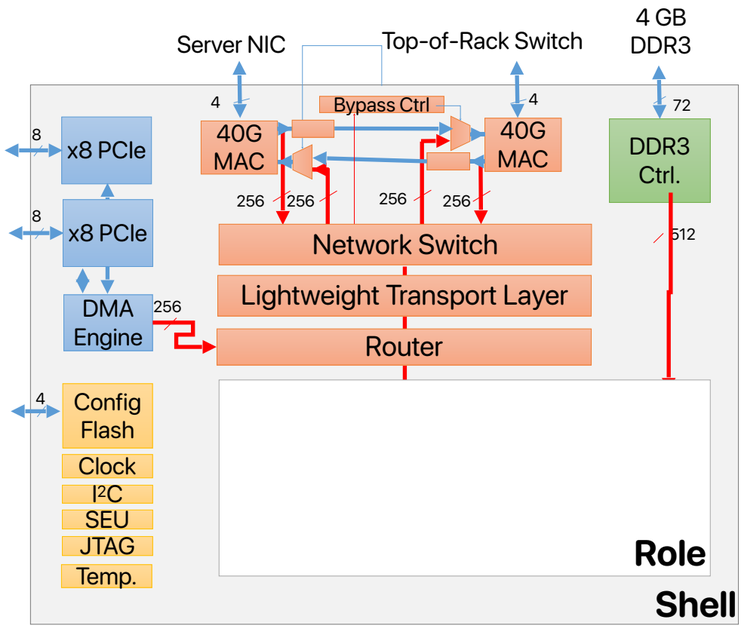
Adrian Caulfield, Eric Chung, Andrew Putnam, Hari Angepat, Jeremy Fowers, Michael Haselman, Stephen Heil, Matt Humphrey, Puneet Kaur, Joo-Young Kim, Daniel Lo, Todd Massengill, Kalin Ovtcharov, Michael Papamichael, Lisa Woods, Sitaram Lanka, Derek Chiou, Doug Burger
Microsoft

Configurable cloud



Gen2 shell

- Foundation for all accelerators
  - Includes PCIe, Networking and DDR IP
  - Common, well tested platform for development
- Lightweight Transport Layer
  - Reliable FPGA-to-FPGA Networking
  - Ack/Nack protocol, retransmit buffers
  - Optimized for lossless network
  - Minimized resource usage

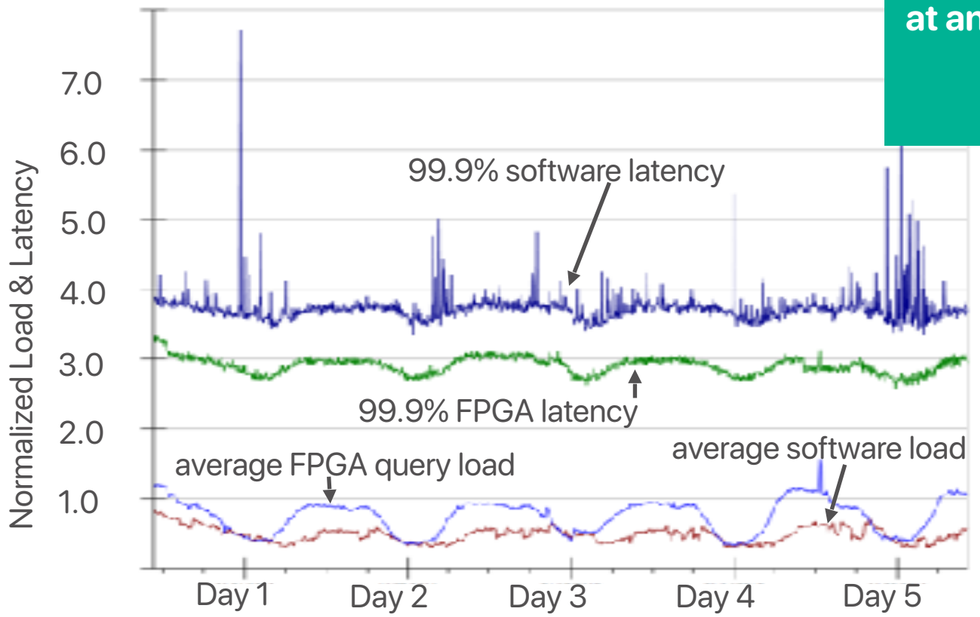


Use cases

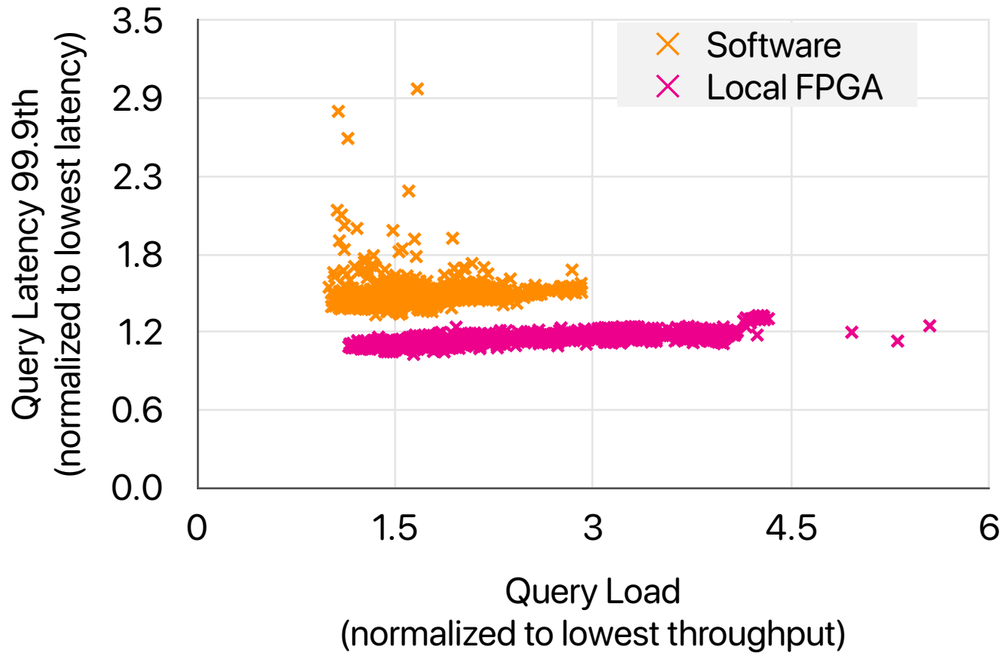
- Local: Great service acceleration
- Infrastructure: Fastest cloud network
- Remote: Reconfigurable app fabric (DNNs)

5 day bed-level latency

- Lower & more consistent 99.9th tail latency
- In production for years



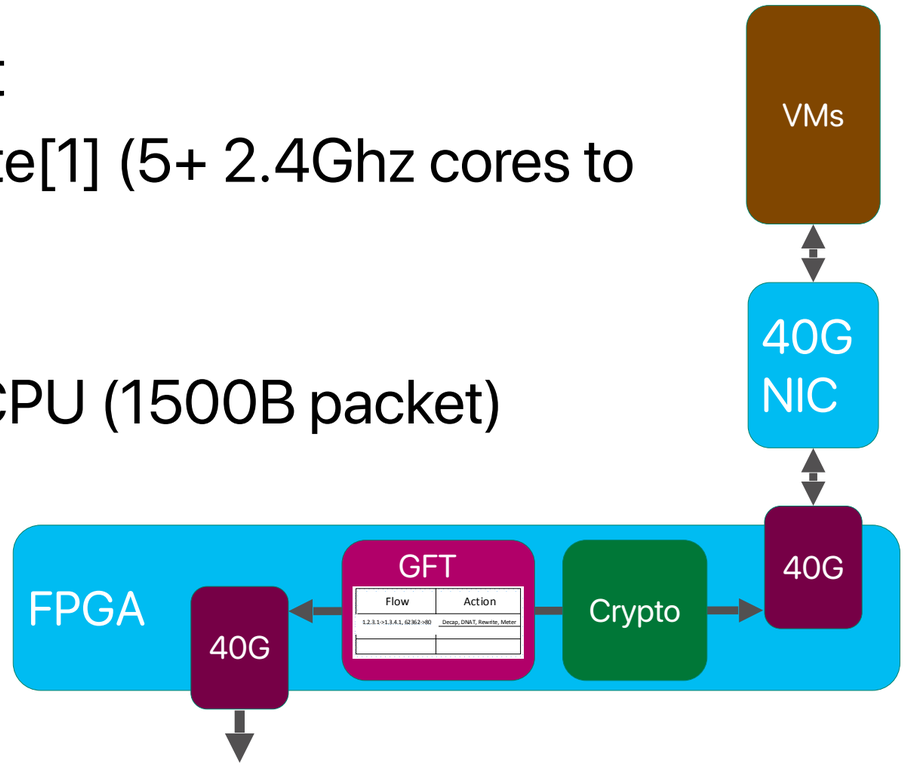
Even at 2× query load, accelerated ranking has lower latency than software **at any load**



Accelerated networking

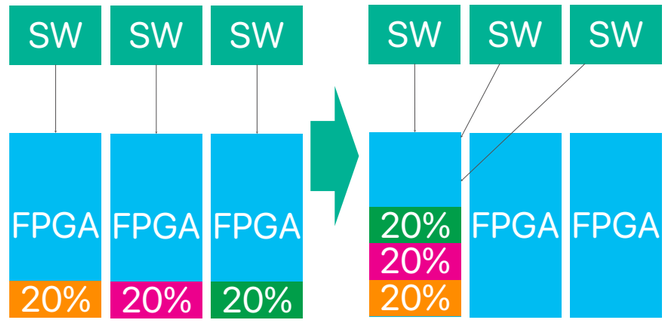
- Software defined networking
  - Generic Flow Table (GFT) rule based packet rewriting
  - 10x latency reduction vs software, CPU load now <1 core
  - 25Gb/s throughput at 25µs latency – the fastest cloud network

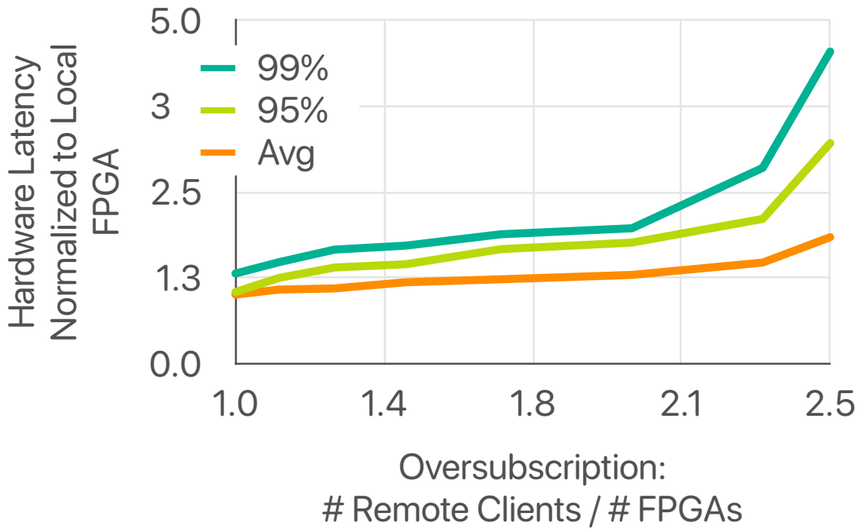
- Capable of 40 Gb line rate encrypt and decrypt
  - On Haswell, AES GCM-128 costs 1.26 cycles/byte[1] (5+ 2.4Ghz cores to sustain 40Gb/s)
  - CBC and other algorithms are more expensive
  - AES CBC-128-SHA1 is 11µs in FPGA vs 4µs on CPU (1500B packet)
  - Higher latency, but significant CPU savings



Shared DNN

- Economics: consolidation
  - Most accelerators have more throughput than a single host requires
  - Share excess capacity, use fewer instances
  - Frees up FPGAs for other use services
- DNN accelerator
  - Sustains 2.5x busy clients in microbenchmark, before queuing delay drives latency up





3:00

What’s your imagination regarding computer architecture and programing 10 years later?

Architects' Bubble

CPU

MEMORY

STORAGE

Ld/St

DRAM

FILE I/O

HDD

SPECIALIZED CORES

DRAM Layers

Logic Layer

Substrate

PERSISTENT MEM

MEMORY WITH LOGIC

SSD

SMART SSD

SMART NIC

BILLIONS OF EDGE DEVICES

Conclusion

- Future computer architecture must be “data-centric”
  - Roofline model — determine what’s the right to do for better performance
    - Compute-intensive — hardware accelerators
    - Memory-bound — near-data processing
  - Minimizing the data movement overhead — larger bandwidth, avoid redundant data transfer
  - Maximizing the hardware utilization — you need to feed data fast enough
- Programming is always an issue
  - Future programming is difficult — computation may be on different parts
  - Carefully gauging the overhead of migrating/synchronizing tasks among different processing units

Next week — you!

Electrical
Computer Science
Engineering

277

つづく