Co-Design of Deep Neural Nets And NN Accelerators

Kurt Keutzer (EECS, UC Berkeley, and Deepscale)
and grad students and post-docs
Alon Amid, Zhen Dong, Amir Gholami, Suresh Krishna,
Bichen Wu, Yang You, Xiangyu Yue, Tianjun Zhang, Sichen Zhao

and an army of Masters and undergraduate researchers
Ravi Krishna,, Aniruddha Nrusimha,
Bernie Wang, Yifan Yang (Tsinghua), Xuanyu Zhou

with fellow faculty, Joey Gonzalez and Mike Mahoney,
and also Krste Asanovic, Jim Demmel and Sanjit Seshia

and as well as Peter Vajda (and others) at Facebook, Kiseok Kwon at Samsung,
Michaela Blott and Kees Vissers (and others) at Xilinx, and Liang Ma and Luciano Lavagno

Accelerators? Who Needs Them?

SnapDragon 835 (→ 845 → 855)

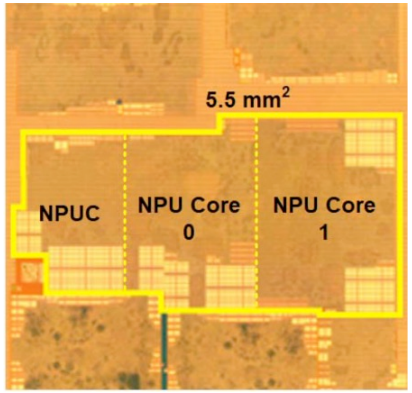
~3.23 – 4 MOPS/mW (835)

11– 16.6 GFLOPs SGEMM (835)

2.5K – 30K X increase in MOPs/mW
TOPS/W

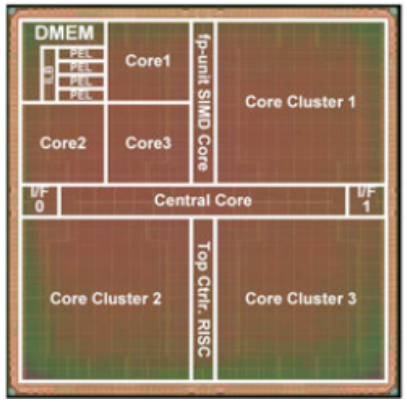
11,500 MOPS/mW

7.1 An 11.5TOPS/W 1024-MAC Butterfly Structure Dual-Core Sparsity-Aware Neural Processing Unit in 8nm Flagship Mobile SoC



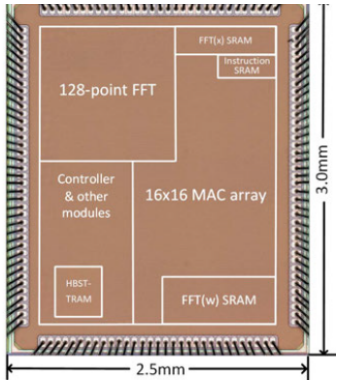
25,300 MOPS/mW

7.7 LNPU: A 25.3TFLOPS/W Sparse Deep-Neural-Network Learning Processor with Fine-Grained Mixed Precision of FP8-FP16

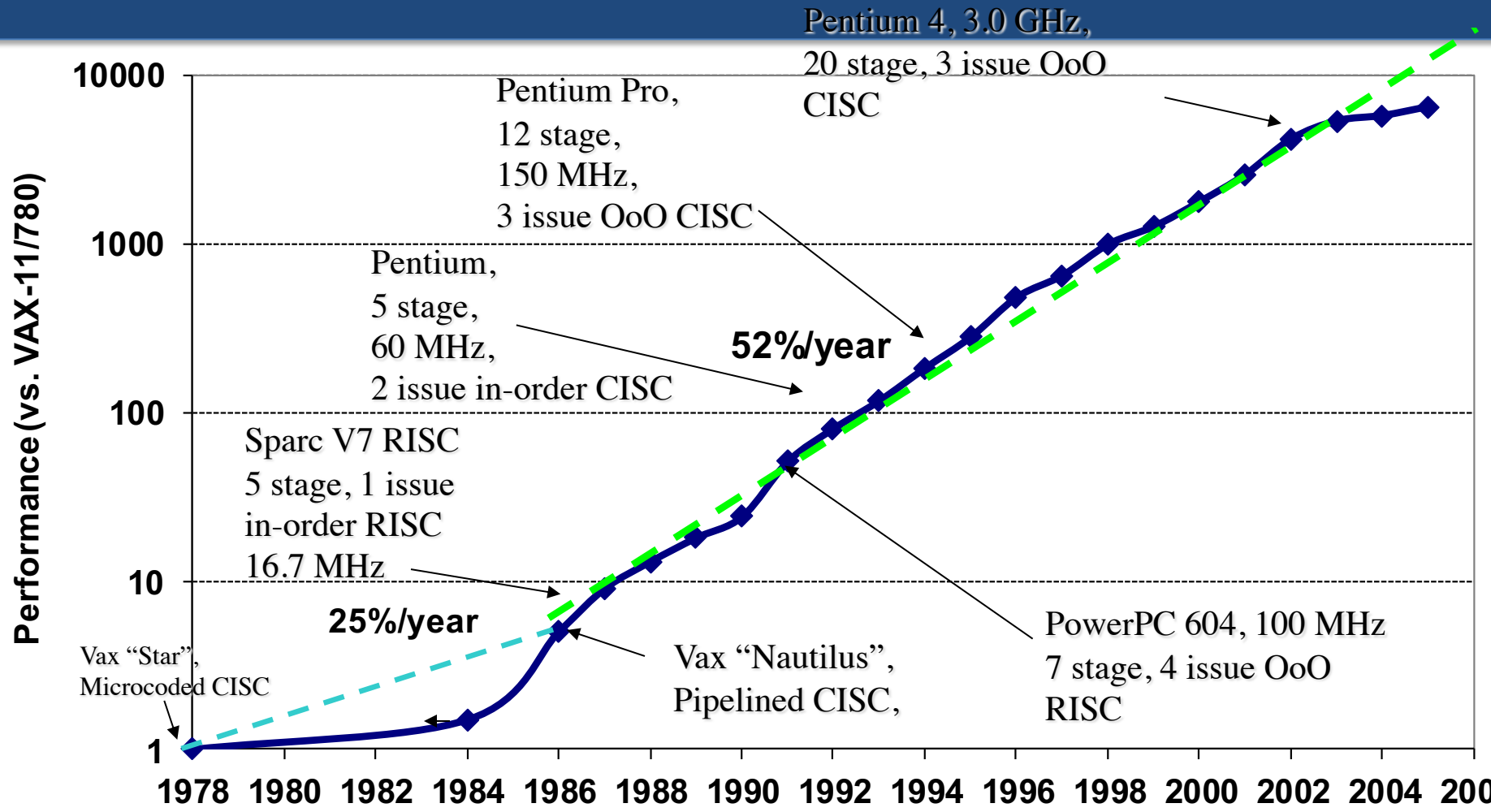


140,300 MOPS/mW

7.5 A 65nm 0.39-to-140.3TOPS/W 1-to-12b Unified Neural-Network Processor Using Block-Circulant-Enabled Transpose-Domain Acceleration with 8.1× Higher TOPS/mm² and 6T HBST-TRAM-Based 2D Data-Reuse Architecture



Why Now?
The Power Wall

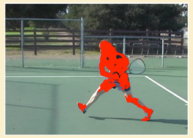


From Hennessy and Patterson, *Computer Architecture: A Quantitative Approach*, 4th edition, Sept. 15, 2006

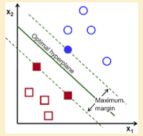
Also: PALLAS Group Machine Learning 2007-2012

Contour Detection

Object Detection

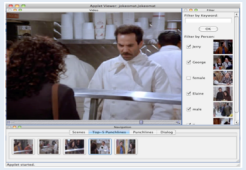


Optical Flow



Support Vector Machines

Computer Vision and Core ML

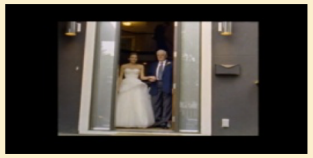


Speaker Diarization

Call-center Sentiment Analysis

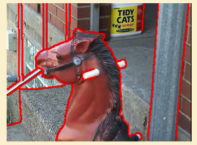
Speech Recognition

Audio Analysis



Video Event Detection

Music Recommendation



Video Segmentation

Multimedia

- Accelerated (10x – 55x) a broad variety of vision, audio, and multimedia problems
- Published in top venues: ICCV, ECCV, CVPR, InterSpeech, ICMR etc.

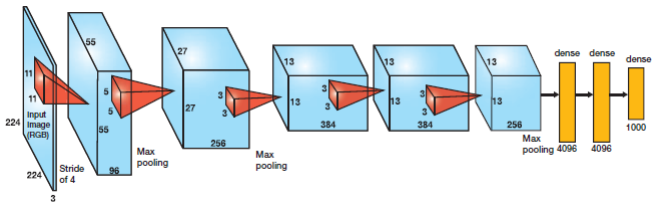
20 Selected Machine Learning Algorithms We Employed

- Computer vision
  - Convolution
  - K-means
  - Mean shift
  - Agglomerative algorithms
  - Vector distance
  - Histogram accumulation
  - Hough transform
  - Eigen decomposition
  - Feature matching
  - Support Vector machines
- Speech recognition and audio analysis
  - Convolution
  - K-means
  - Agglomerative hierarchical modeling
  - Orthogonal transformations
  - Gaussian Mixture models
  - Weighted-finite state transducers
  - Hidden-Markov-models
  - Dynamic Bayesian networks
  - Expectation maximization

- Distilled these algorithms down to their “computational patterns” or “dwarfs”
- Accelerated those computational patterns
- GPUs did well on some (e.g. SVM) not so well on others (e.g. HMM)

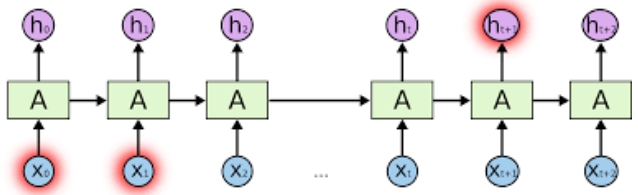
ML Algorithms were Displaced by *a Single DNN*! Now it's really clear what to accelerate

- Computer vision
  - Convolution
  - K-means
  - Mean shift
  - Agglomerative algorithms
  - Vector distance
  - Histogram accumulation
  - Hough transform
  - Eigen decomposition
  - Feature matching
  - Support Vector machines



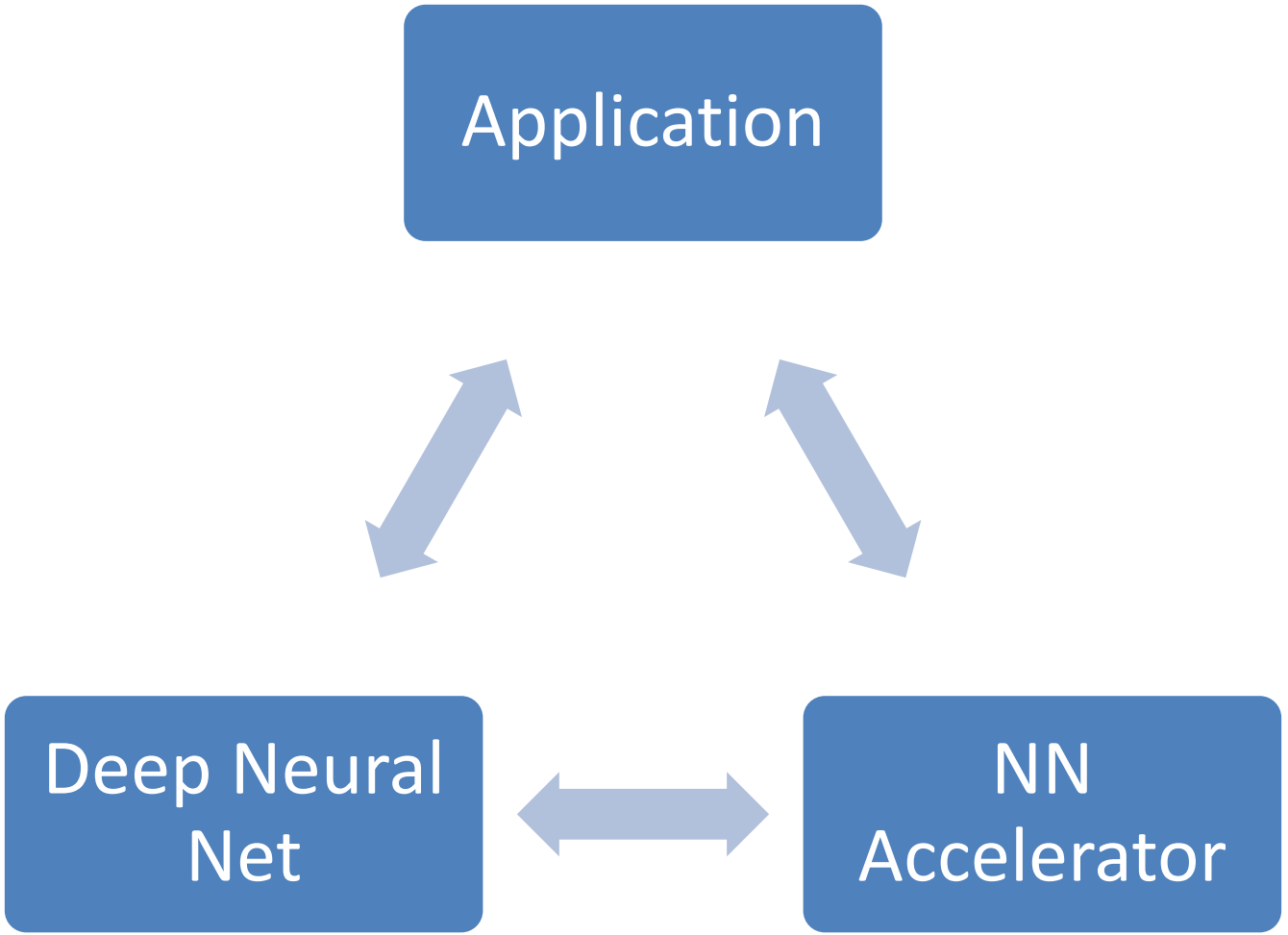
Convolutional Neural Nets

- Speech recognition and audio analysis
  - Convolution
  - K-means
  - Agglomerative hierarchical modeling
  - Orthogonal transformations
  - Gaussian Mixture models
  - Hidden-markov models
  - Dynamic Bayesian network
  - Expectation maximization



Recurrent Neural Nets

Co-Design of NN Accelerators: The Ideal



Co-Design: The Reality

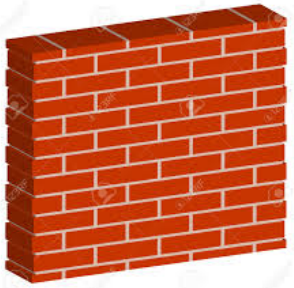
Application

Misrepresentative application data
- e.g. ImageNet

Only gross constraint
- power budget
- cost
- form factor

Deep Neural Net

Outdated information exchange – if any at all

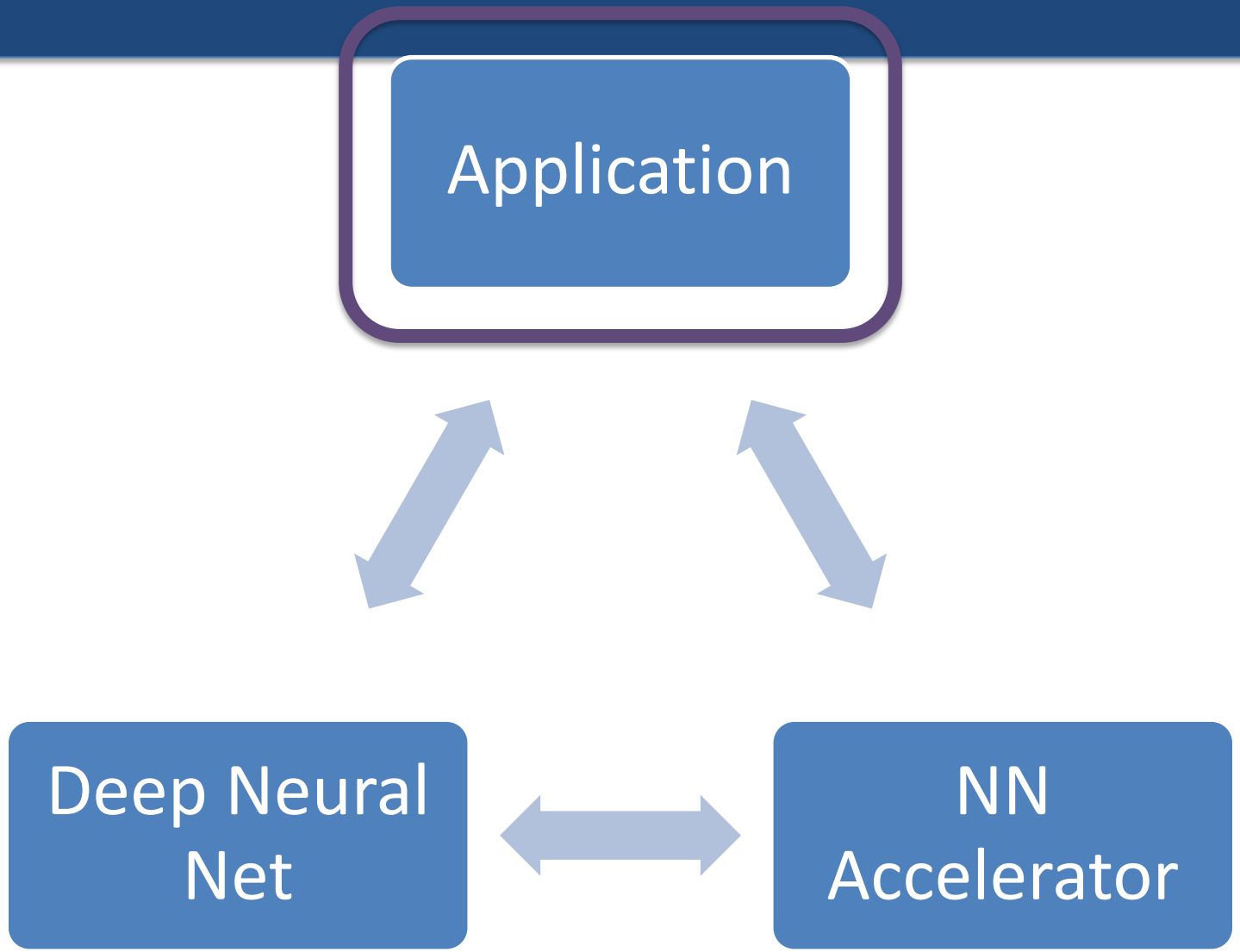


NN Accelerator

- counting FLOPs/MACS
- counting model parameters

- Datasets: MNIST, CIFAR
- DNN: AlexNet, VGG

Let's Do Better!



Outline

- Applications their characteristics, and targets

- Determining key NN Accelerator architectural elements

- How much further can we improve NN accelerators with co-design?

AI Chip Landscape

basicmi.github.io/AI-chip

Tech Giants/Systems

Google
Microsoft
aws
IBM
facebook
Apple
Tesla
HUAWEI
Baidu 百度
Alibaba Group 阿里巴巴集团
FUJITSU
NOKIA
TOSHIBA
Hewlett Packard Enterprise
DELL

IC Vender/Fabless

intel
SAMSUNG
NVIDIA
QUALCOMM
AMD
NXP
ST
XILINX
MEDIATEK
BROADCOM
MARVELL
Rockchip 瑞芯微电子

IP/Design Service

arm
SYNOPSYS
Imagination
cadence
CEVA
VeriSilicon
alchip
GUC
FARADAY
eSilicon

Startup in China

Cambricon 寒武纪科技
地平线 Horizon Robotics
BITMAIN
intellifusion 云天励飞
ChipIntelli
Think Force
Canaan
云知声 Unisound
AISPEECH 思必驰 专注人性化的智能语音
Rokid
肇观电子 NextVPU
Enflame

Startup Worldwide

cerebras
Wave Computing
Graphcore
habana
SambaNova SYSTEMS
thinci
LIGHTELLIGENCE
HAILO Empowering Intelligence
KALRAY
Tenstorrent
MYTHIC
Preferred Networks
brainchip
PEZY Computing
GREENWAVES TECHNOLOGIES
AIMOTIVE
KONIKU
Tachyum
flexlogix
SYNTIANT
gyrfalcon technology
NOVUMIND

more on https://basicmi.github.io/AI-Chip/

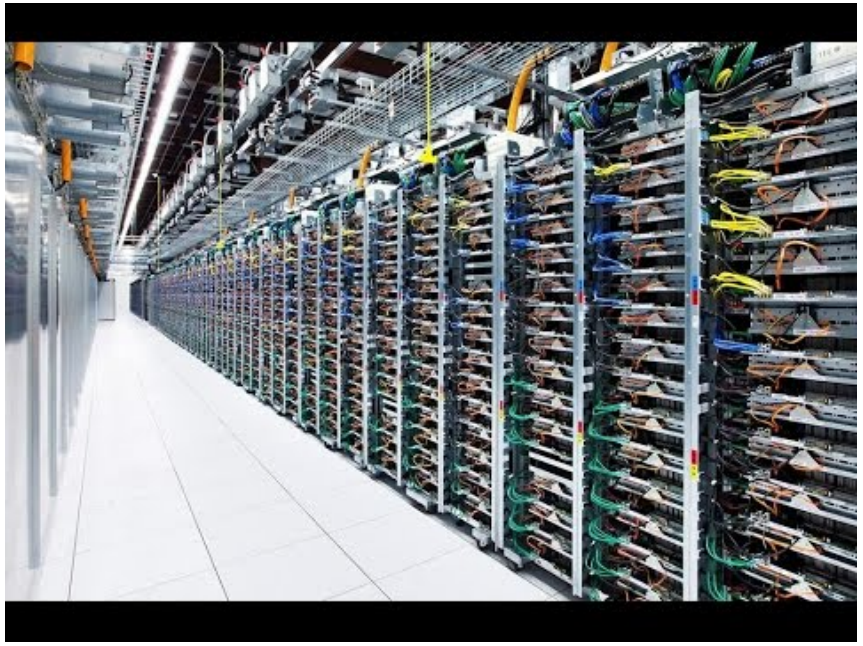
Compiler

TensorFlow
PyTorch
NVIDIA TensorRT
tvm
nGraph Compiler stack (Beta)
plaidML

Benchmarks

MLPerf
AI - Benchmark
AI Matrix.

Three Broad Targets for Deployment



DataCenter

IOT/Edge
AIOT

Mobile Cell Phone
(IP Block)

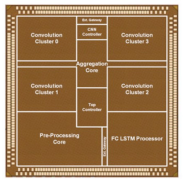
Three Broad Classes of Accelerators

Training/Inference in the Cloud

- Training + Inference
- 10,000s PEs
- Area: 300 -- 800 mm^2
- 80--300 Watts
- 60-150s TOPS
- Clock rate: 700MHz – 1.6GHz
- 10s MB on-chip memory
- Typical batch size: 10s – 100s
- High/complete connectivity PEs
- 8-32 bit precision (floating point)
- Example: TPU 1-3, Graphcore

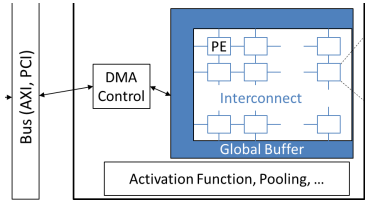
Inference at the Edge: Standalone Chip

- Only Inference
- 16 -1000s of PEs
- Area: 1s-10s mm^2
- 1 – 10,000 mWatts
- 100s-1000 GOPS
- Clock rate: 25-400 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- mixed connectivity among PE
- 1-16 bit precision (fixed point)
- Example: GreenWaves, ZU3 + Gyrfalcon Lightspeeur, Synetgy



Inference at the Edge: IP Block/System-in-Package

- Only Inference
- 16-64 PEs
- Area: 1s mm^2
- 10's-100s mWatts
- 10s-5000s GOPS
- Clock rate: 600 – 1000 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- Local (mesh) connectivity
- 1-16 bit precision (fixed point)
- Example: Apple NPU, Tensilica DNA 100, Squeezelerator



Cloud Workloads at Google
Dave Patterson UC Berkeley 10/2019

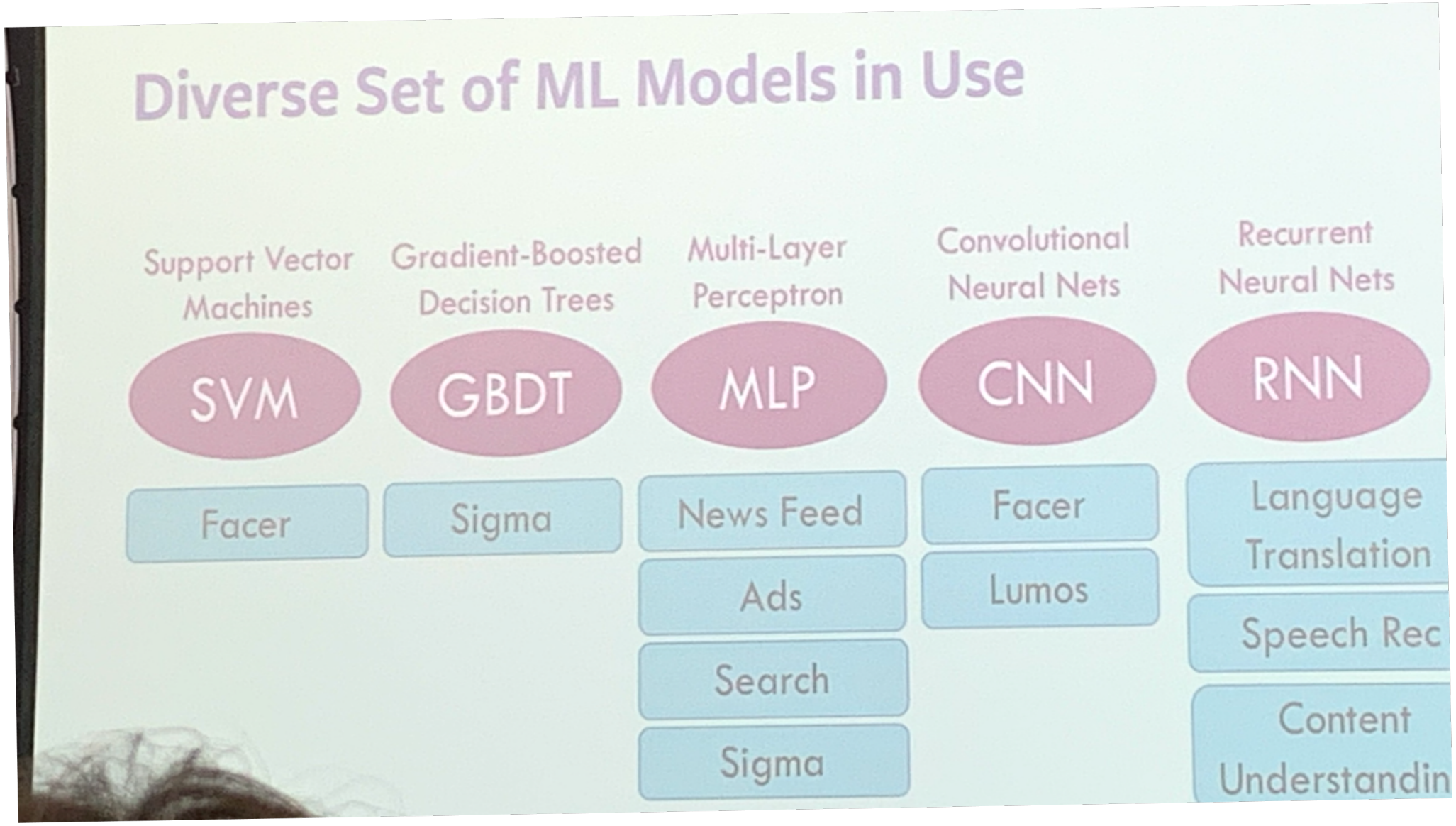
Model Popularity

DNN Model	TPUv1 July 2016 (Inference)	TPUv3 April 2019 (Training)
MLP	61%	27%
RNN	29%	21%
CNN	5%	24%
Transformer	–	21%

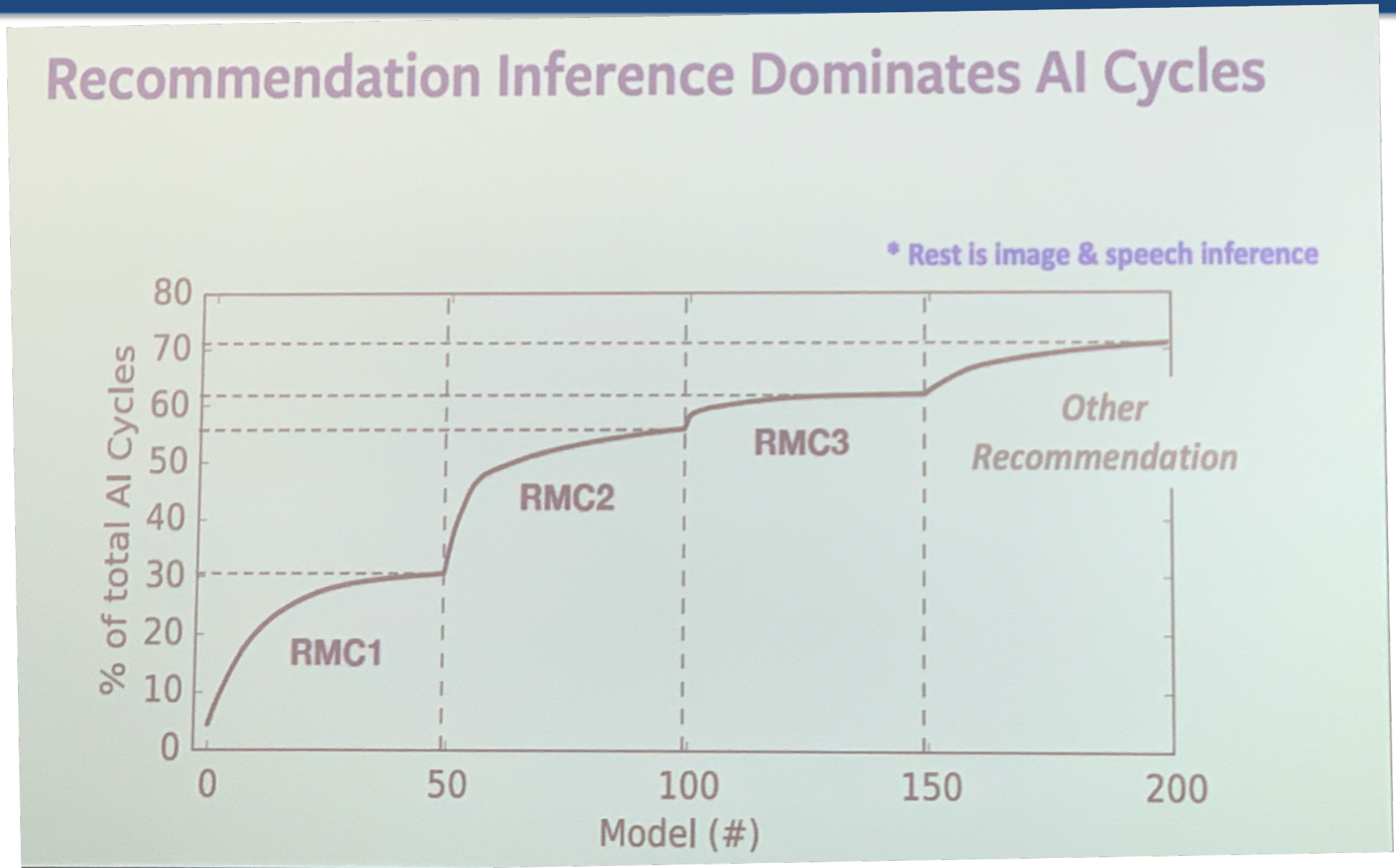
- Inference in 2016 vs Training in 2019 on TPUs in Google Datacenters
- Transformer for same tasks as RNNs (e.g., translation) but considerably faster since it's parallel while RNNs have sequential dependencies
  - Transformer published same year TPUv2 deployed
  - Layers of Transformer are a mix of MLPs and attention layers [Bah14]
  - Success of recent Transformer model highlights the programmability of TPUs

55

Cloud Workloads at FB
Carole-Jean Wu, FB AI Systems Faculty Summit 9//2019



Cloud Workloads at FB
Carole-Jean Wu, FB AI Systems Faculty Summit 9//2019



GPU as DL Accelerators

Product	K40	M40	Tesla P100	Tesla V100	T4
GPU	GK180 Kepler	GM200 Maxwell	GP100 Pascal	GV100 Volta	TU104
Year	2013	2015	2016	Early 2018	Late 2018
Streaming Multiprocessors	15	24	56	80	40
FP32 cores/SM	192	128	64	64	64 CUDA cores
FP32 cores/GPU	2880	3072	3584	5120	2560 CUDA Cores
Tensor cores/SM	N/A	N/A	N/A	8	8
Tensor cores/GPU	N/A	N/A	N/A	640	320
Peak TOPS FP32	5	6.8	10.6	15.7	8.1
Peak TOPS Tensor	N/A	N/A	N/A	125 fp16xfp16→fp32	130 TOPS INT8, 260 TOPS INT4
TDP (W)	235 W	250W	300W	300W	70W
TOPS/W	***0.02 TFLOPS/W fp32***	***0.027 TFLOPS/W fp32***	***0.035 TFLOPS/W fp32***	***0.42 TFLOPS/W tensor fp16/32***	***1.85 TOPS/W INT8***
Shared memory/SM (L1 cache)	16kB-48kB	96kB	64kB	Up to 96kB	96kB
RegFile/SM (per GPU)	256kB (3840kB)	256kB (6144kB)	256kB	256kB	256kB
L2 Cache	1536kB	3072kB	4096kB	6144kB	4096kB

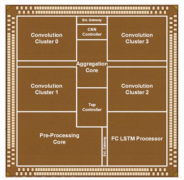
Three Broad Classes of Accelerators

Training/Inference in the Cloud

- Training + Inference
- 10,000s PEs
- Area: 300 -- 800 mm^2
- 80--300 Watts
- 60-150s TOPS
- Clock rate: 700MHz – 1.6GHz
- 10s MB on-chip memory
- Typical batch size: 10s – 100s
- High/complete connectivity PEs
- 8-32 bit precision (floating point)
- Example: TPU 1-3, Graphcore

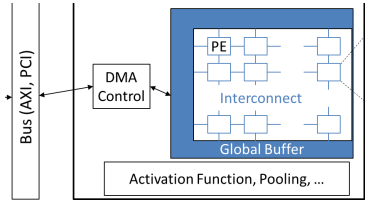
Inference at the Edge: Standalone Chip

- Only Inference
- 16 -1000s of PEs
- Area: 1s-10s mm^2
- 1 – 10,000 mWatts
- 100s-1000 GOPS
- Clock rate: 25-400 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- mixed connectivity among PE
- 1-16 bit precision (fixed point)
- Example: GreenWaves, ZU3 + Gyrfalcon Lightspeeur, Synetgy

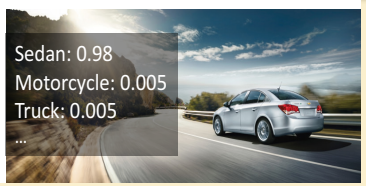


Inference at the Edge: IP Block/System-in-Package

- Only Inference
- 16-64 PEs
- Area: 1s mm^2
- 10's-100s mWatts
- 10s-5000s GOPS
- Clock rate: 600 – 1000 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- Local (mesh) connectivity
- 1-16 bit precision (fixed point)
- Example: Apple NPU, Tensilica DNA 100, Squeezelerator

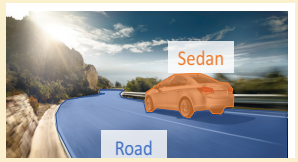


Most Popular ML/DL Applications at the Edge and their DNNs



Classification

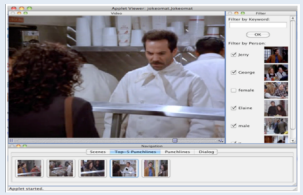
Object Detection



Semantic Segmentation

Computer Vision: CNNs, ConvNets

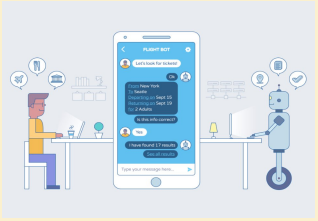
Speech Enhancement



Speaker Diarization

Speech Recognition

Audio/Speech: LSTMs, RNN



Chat Bots

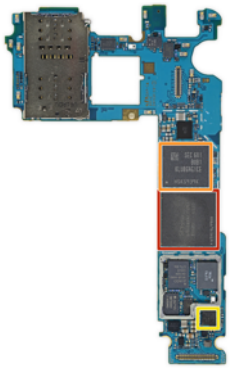
Recommendation Systems

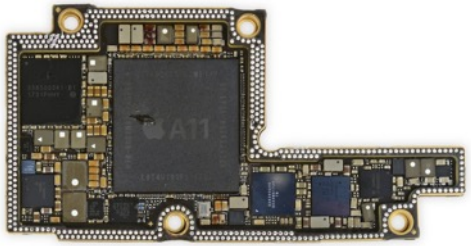
Call-center Sentiment Analysis

Natural Language transformer networks

- Models characteristics (model size, computations, Arithmetic Intensity) vary widely from application area to application area

Mobile Devices Bring Constraints





Power (Watts)

- Convenient and economical **packaging** limits how much power our mobile devices can dissipate
- ***2-5W*** max seems common among mobile handsets
- IP Blocks will have much stricter constraints

Energy (Joules) = power * time

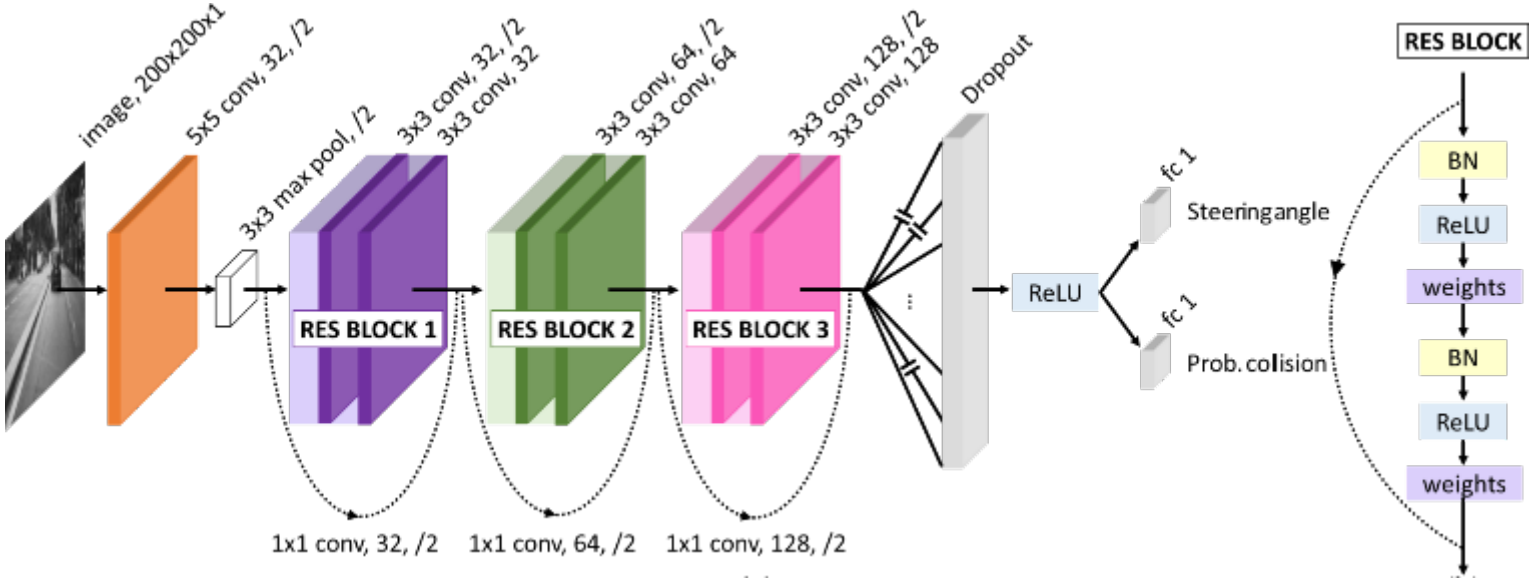
- **Battery life** limits the total energy that our mobile devices can use
- iPhoneX battery ***10.35 WHours***

Applications may bring further constraints on accuracy and latency

https://www.macrumors.com/2017/11/03/iphone-x-teardown-ifixit/

	Snapdragon 845	Snapdragon 855
Timeframe, Phone, Process node	Feb. 2018, Galaxy S9, Samsung 10nm [2]	Q1 2019, Galaxy S10, TSMC 7nm [6]
Die area	Entire chip 95mm^2 [2]	Entire chip 73mm^2 [6]
CPU cores	4x A75 @ 2.8GHz, 4x A55 @ 1.8GHz	4x A76+ @ 2.4-2.8GHz, 4x A55 @ 1.8GHz
CPU FP Perf.	2x 64-bit pipes [9], can do 2x fp32 or 8x int8 each ➔ ~90 GFLOPS for 4 cores	2x 128-bit pipes, can do 4x fp32 or 16x int8 each ➔ 154-180 GFLOPS for 4 cores
Actual CPU perf, Geekbench 4, SGEMM	~66 GFLOPS / 4 cores [5]	Qualcomm quotes +35% FP perf over 845
CPU int8 perf / ARM dot product	Int8 dot product with int32 accumulation, 4-cycle MAC 16 ops/clock (8 macs) ➔ 153-180 GOPS int8 Shares Neon vector pipe with FP unit?	Int8 dot product with int32 accumulation, 1-cycle MAC 64 ops/clock (16 macs) ➔ 614-717 GOPS int8 Shares Neon vector pipe with FP unit?
GPU	Adreno 630 @ 710MHz, dual-core [3]	Adreno 640 @ 585MHz, tri-core [3]
GPU Perf.	727 GFLOPS fp32 [3]	955 GFLOPS fp32, 1853 GFLOPS fp16 [3]
NPU	Hexagon 685 DSP w/ vector extensions + NPE Neural Processing Engine 1.8 TOPS int8 [8]	Hexagon 690 DSP + TensorAccel DSP Est. 1.5 int8 TOPS TensorAccel Est. 32x32 array, 2 int8 macs each/clock ➔ 4.4 int8 TOPS + ~1.5 int8 TOPS DSP
Android NNAPI support [7]	Int8 on DSP, fp16 on and fp32 on GPU ??	Same as 845

IOT END: Standalone Chip: GreenWaves Gap8
DRONET: RESNET based Autonous Drone



- Developed by UZH and ETH-Z
- Autonomously follow a road/corridor and avoid collision
- ***Up to 18 Frames Per Second*** at maximum frequency
- @1.0V, FC: 50MHz, Cluster: 100MHz ➔ **6.5fps 40mW**

GREENWAVES TECHNOLOGIES

Courtesy, Eric Flamand, CTO

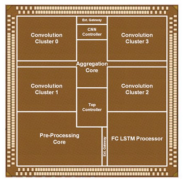
Three Broad Classes of Accelerators

Training/Inference in the Cloud

- Training + Inference
- 10,000s PEs
- Area: 300 -- 800 mm^2
- 80--300 Watts
- 60-150s TOPS
- Clock rate: 700MHz – 1.6GHz
- 10s MB on-chip memory
- Typical batch size: 10s – 100s
- High/complete connectivity PEs
- 8-32 bit precision (floating point)
- Example: GPUs, TPU 1-3, Graphcore

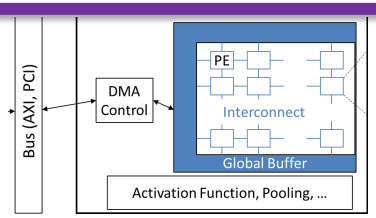
Inference at the Edge: Standalone Chip

- Only Inference
- 16 -1000s of PEs
- Area: 1s-10s mm^2
- 1 – 10,000 mWatts
- 100s-1000 GOPS
- Clock rate: 25-400 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- mixed connectivity among PE
- 1-16 bit precision (fixed point)
- Example: GreenWaves, ZU3 + Gyrfalcon Lightspeeur, Synetgy



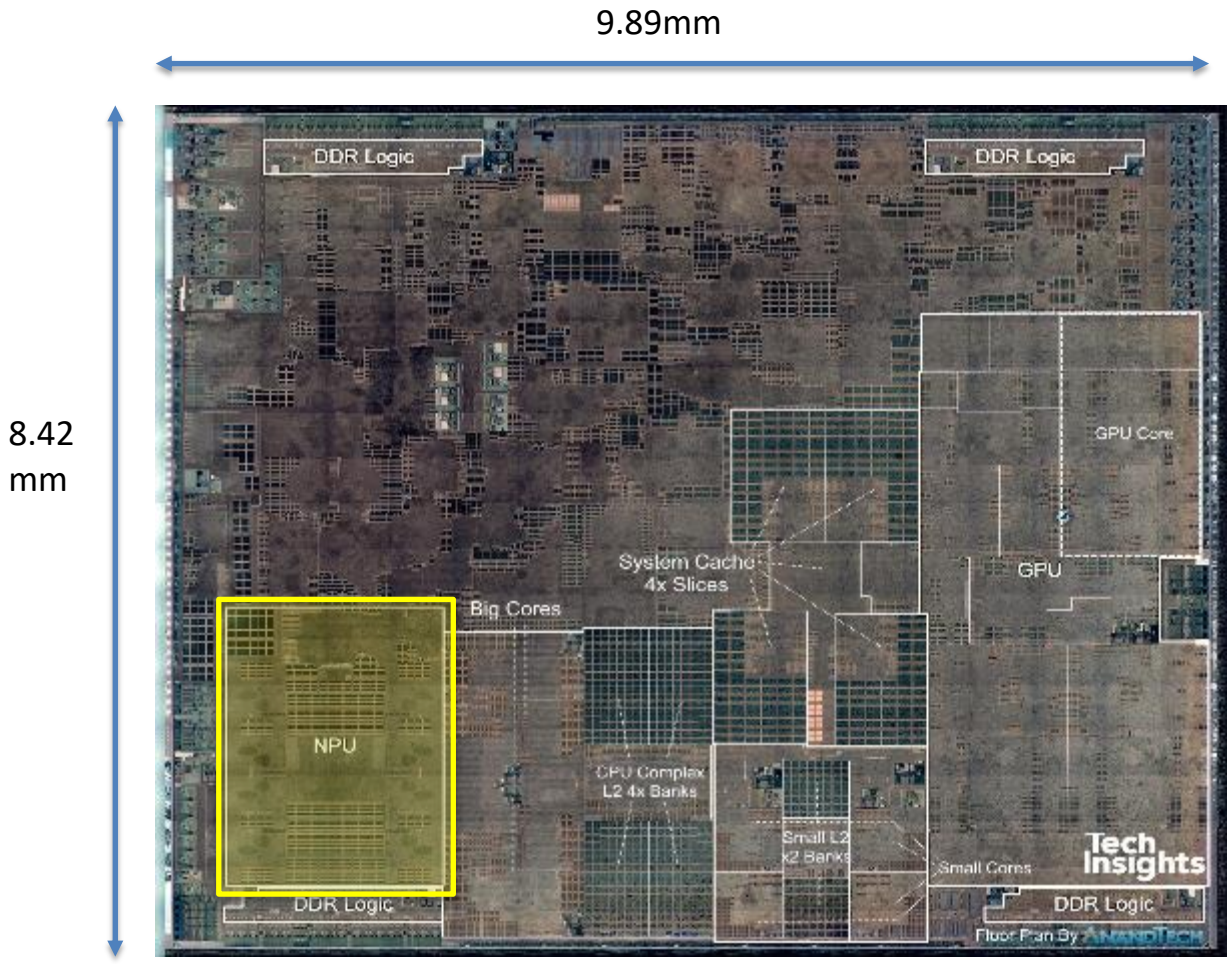
Inference at the Edge: IP Block/System-in-Package

- Only Inference
- 16-64 PEs
- Area: 1s mm^2
- 10's-100s mWatts
- 10s-5000s GOPS
- Clock rate: 600 – 1000 MHz
- 100s KB on-chip memory
- Typical batch size: 1
- Local (mesh) connectivity
- 1-16 bit precision (fixed point)
- Example: Apple NPU, Tensilica DNA 100, Squeezelerator



Embedded IP Block in Mobile Phone
Apple A12 Bionic NN Engine

Apple A12 Bionic
~6.9B transistors
TSMC 7nm, ~83 mm^2
5 TOPS peak
NN Engine:
8 cores, int8
NPU ~5.8 mm^2



https://www.anandtech.com/show/13392/the-iphone-xs-xs-max-review-unveiling-the-silicon-secrets/2
https://en.wikichip.org/wiki/apple/ax/a12

Like to have a NN Accelerator Architectural Family That is Useful Across Wide Range of Mobile Apps

Average Power Dissipation

Battery Life

11 W
2.4 Hours

2.5 W
8.4 Hours

1-1000 mW
8-240 Hours

15 inch Macbook Pro |
76Wh = 273.6kJ
Apple: 10h = 7.6 W

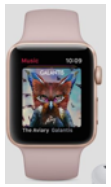
13 inch Macbook Air | 54Wh = 194.4kJ
Apple: 12h = 4.5 W

Eee PC 1000HE | 49Wh = 176kJ
Asus: 9.5h = 5.2 W

iPad Pro | 41Wh = 147kJ
Apple: 10h use = average 4.1W

Kindle Oasis | 0.91Wh = 3.276kJ
Ebook Friendly: "15days @ 30m/day" = 7.5h @ 0.12 W average

2017 iPhone8 | 6.96Wh = 25kJ
Talk time: 14h = 0.5 W
Video: 13h = 0.54 W



iwatch Series 3 1.07Wh | 3.8kJ
60mW, 18 hours

OZO Digital Pedometer
80µW, 0.72Wh | 1 year

1 Wh = 3.6 kJ

Typical handset

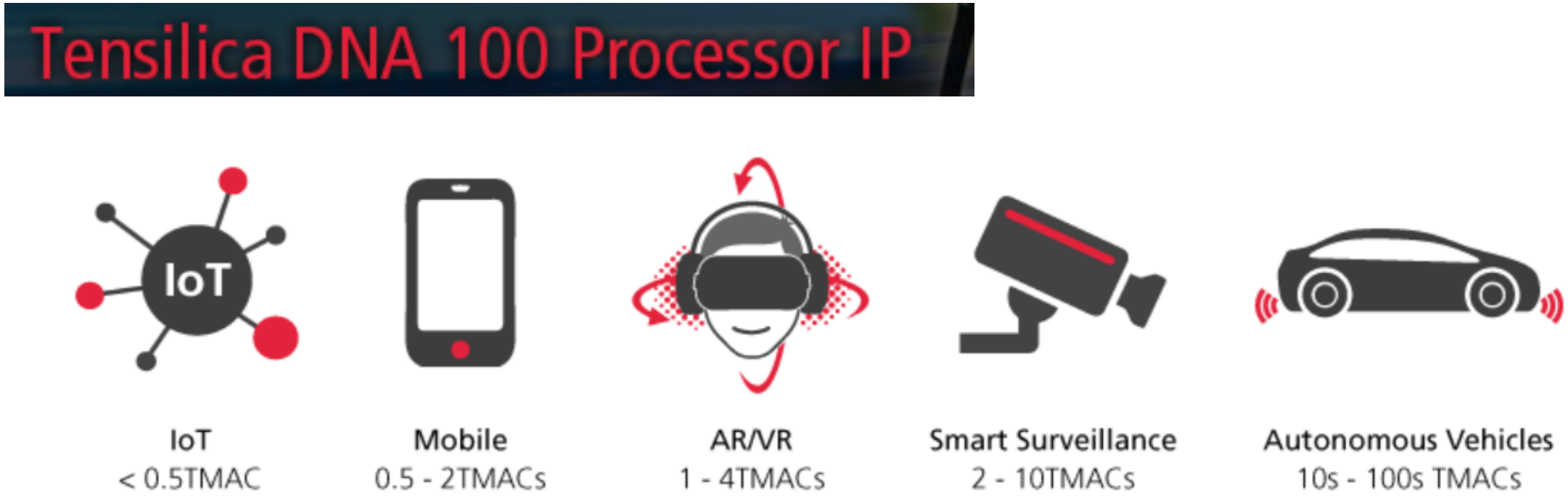
32g, 13cc, 5.5Wh = 19.8 kJ

Typical usage

5kJ active + 12kJ standby = 1 battery charge

Per Ljung – Nokia, 2012

Architectural Family to Support Broad Range of Inference at the Edge



- "Enabling On-Device AI Across a Wide Range of Inference from 0.5 to 100s of TMACs"

High level NN Accelerator Application Characteristics For Mobile/Embedded Applications

- Characteristics of Deep Neural Nets for embedded computer vision
  - (Only) need to support varieties of Convolution Nets
    - *But do need to recognize their diverse layers!*
  - Moreover, to support a wide range of *accuracies* we will need to support a wide range of size of models

Accuracy requirements important: Precise requirements are application specific

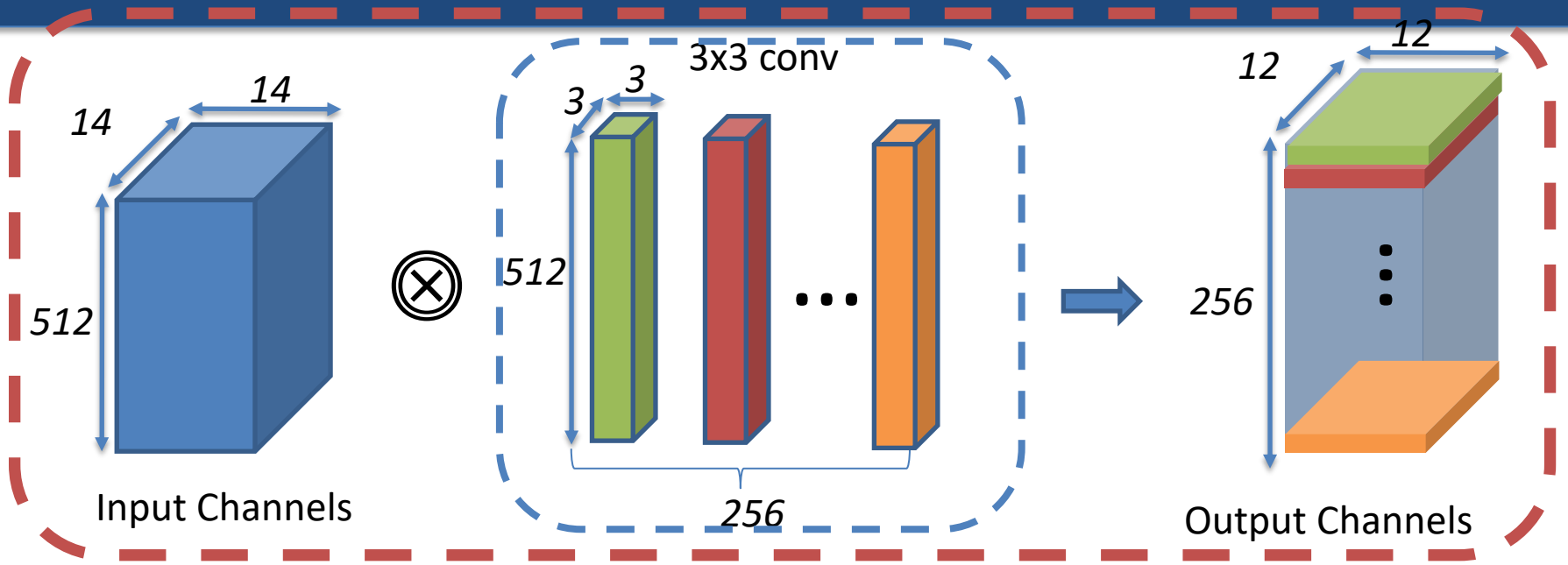
Application

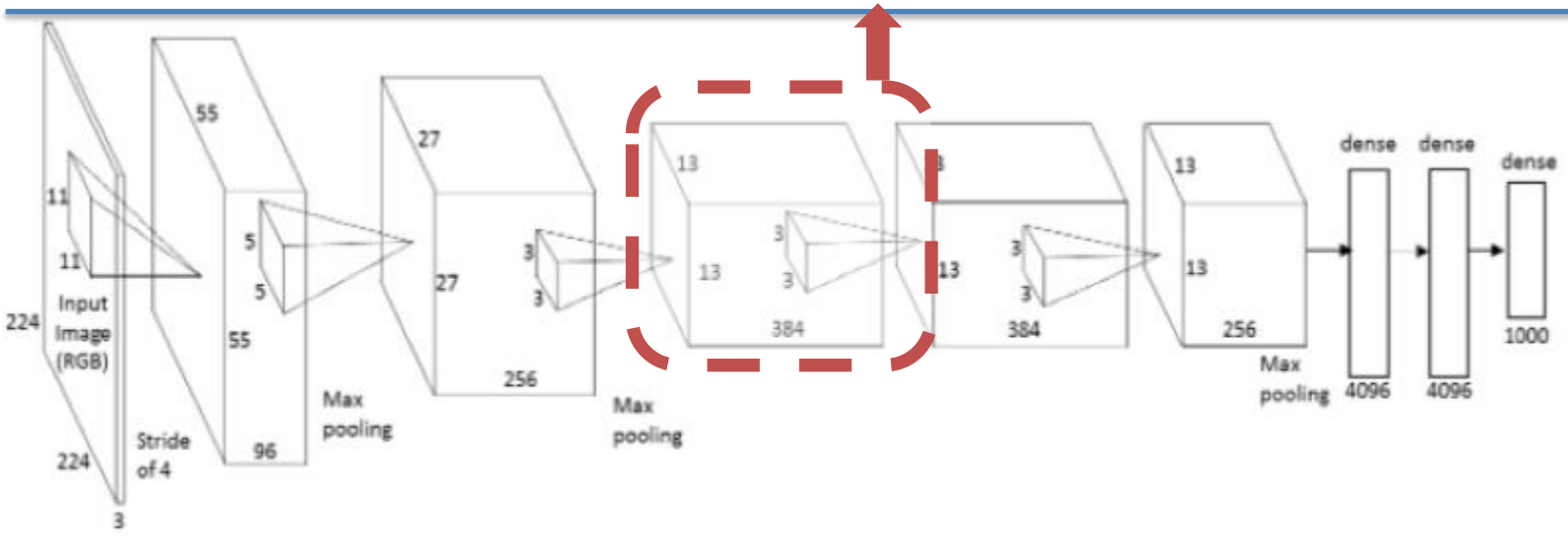
iPhone Dog Identifier Application

Object Detector in Autonomous Vehicle

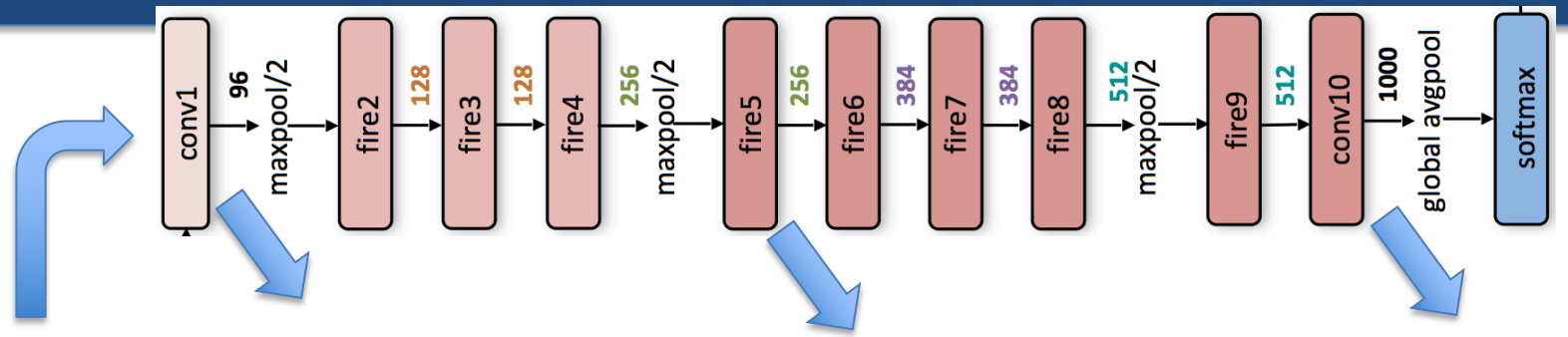
- ***Most overlooked question in Deep Learning: what does accuracy mean?***

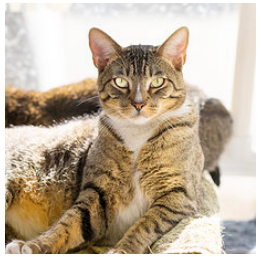
At the Core of a CNN is ...
A Convolution



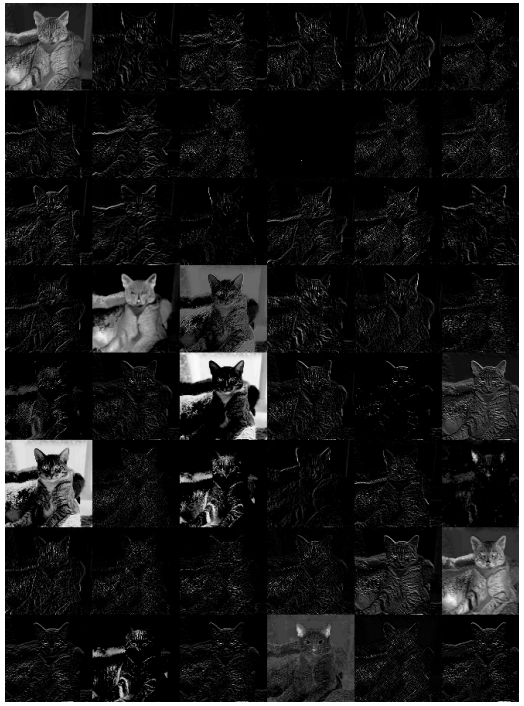


Convolutional Layers in SqueezeNet

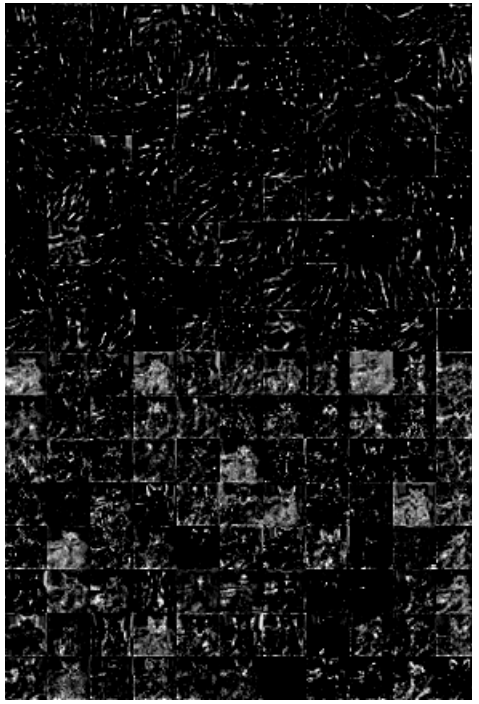




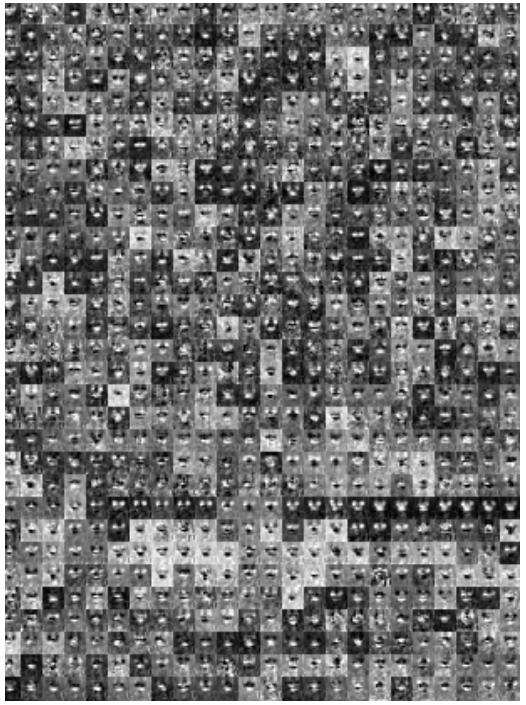
3 channels (RGB)



96 channels



256 channels

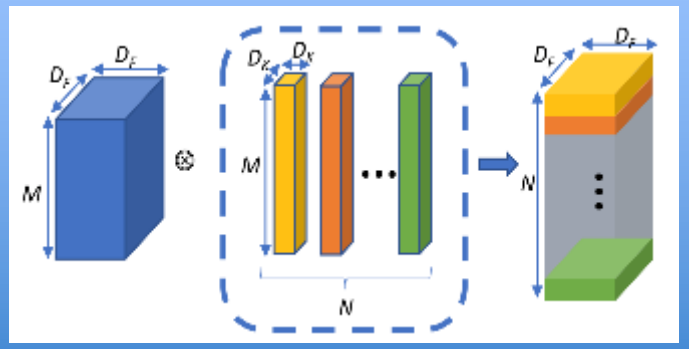


1000 channels

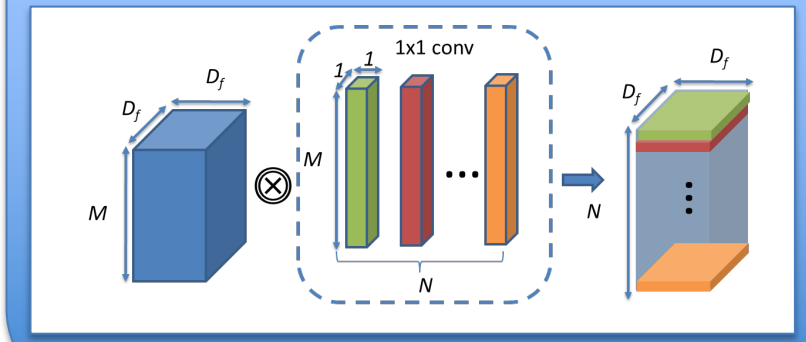
- https://deeplearnjs.org/demos/imagenet/

Actually Many Different Variants
All Have Different Computational Characteristics

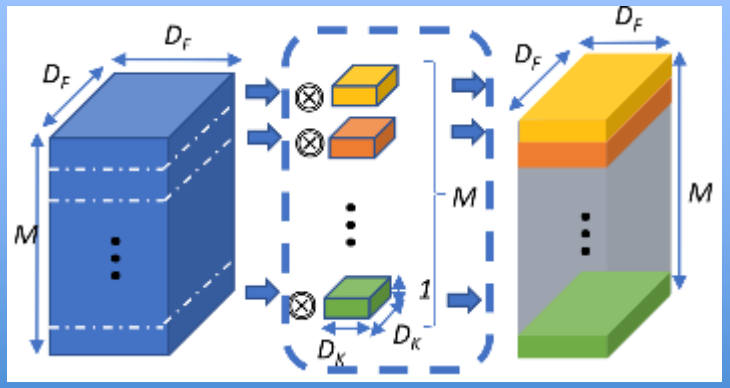
Spatial Convolution
e.g. 3x3



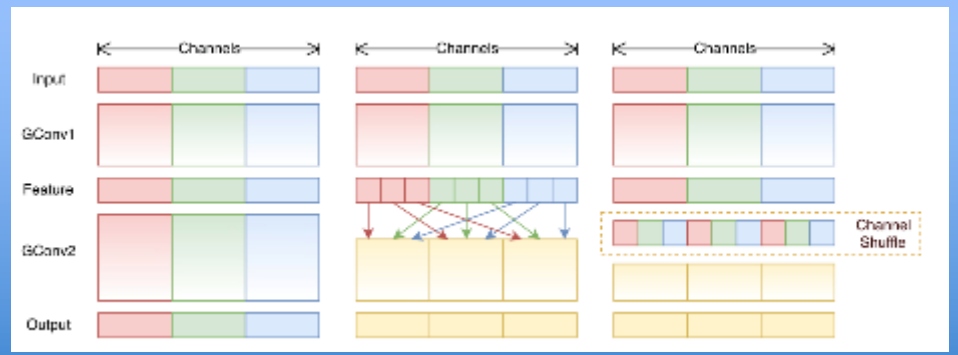
Pointwise Convolution
1x1



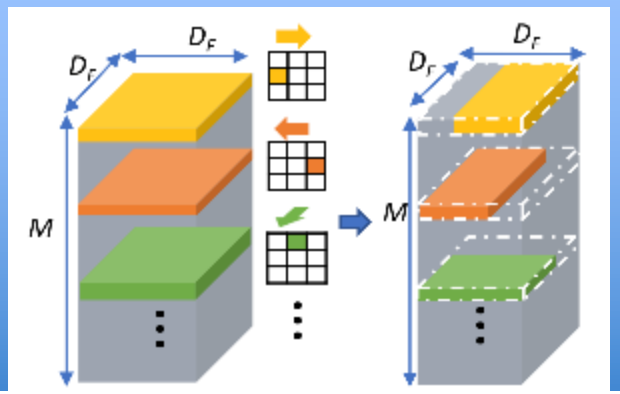
Depthwise Convolution



Channel Shuffle



Shift



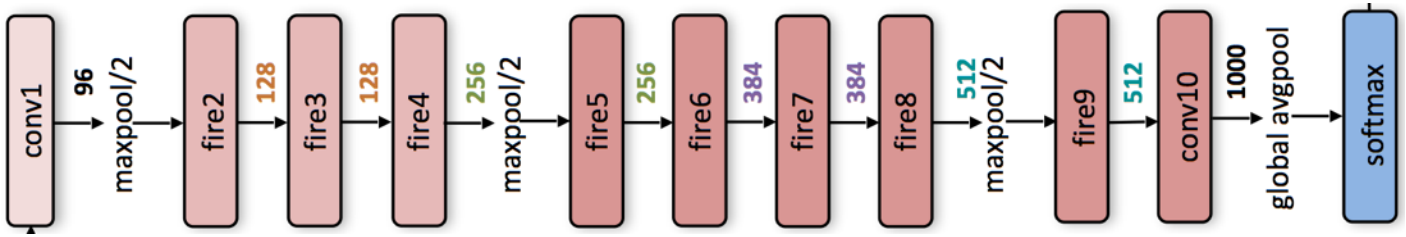
Alon Amid, ... Kurt Keutzer. "Co-Design of Deep Neural Nets and Neural Net Accelerators for Embedded Vision Applications." Invited paper: to appear IBM Journal of Research and Development.

High level NN Accelerator Application Characteristics For Mobile/Embedded Applications

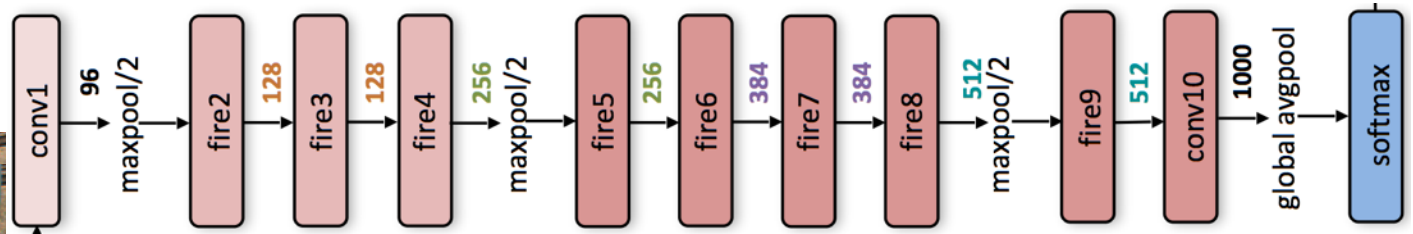
- Characteristics of Deep Neural Nets for embedded computer vision
  - (Only) need to support varieties of Convolution Nets
    - *But do need to recognize their diverse layers!*
  - Moreover, to support a wide range of accuracies we will need to support a wide range of size of models
- Batch size?

Latency Constraint: Natural Batch Size for Embedded/IOT/Mobile is 1

Batch Size of 1



car



<car, car, ... car>

Batch Size of 16, 44 etc.

- Batching up input images allows for more reuse of convolutional filters → better numbers
- However, real-time requirements will almost always dictate minimizing latency → batch size 1

High level NN Accelerator Application Characteristics For Mobile/Embedded Applications

- Characteristics of Deep Neural Nets for embedded computer vision
  - (Only) need to support varieties of Convolution Nets
    - *But do need to recognize their diverse layers!*
  - Moreover, to support a wide range of accuracies we will need to support a wide range of size of models
- Benchmark using batch size of 1
  - Presuming larger batch sizes, as is common, will give unrealistic numbers on FPS etc.

NN Accelerator Application Characteristics For Mobile/Embedded Applications

- Characteristics of Deep Neural Nets for embedded computer vision
  - (Only) need to support varieties of Convolution Nets
    - *But do need to recognize their diverse layers!*
  - Moreoever, to support a wide range of accuracies we will need to support a wide range of size of models
- Presume batch size of 1
  - Presuming larger sizes, as is common, will give unrealistic numbers on OPS/Watt
- Power ranges as low as 1mW and up to 10W
- Capable of providing real time performance for common computer vision kernels (classification, object detection) across:
  - Range of resolutions: CIFAR (32 x 32) up to UHD (2160 x 3840)
  - Range of speeds: 1 Frames per second (FPS) → 60 FPS

Outline

- Applications and their characteristics

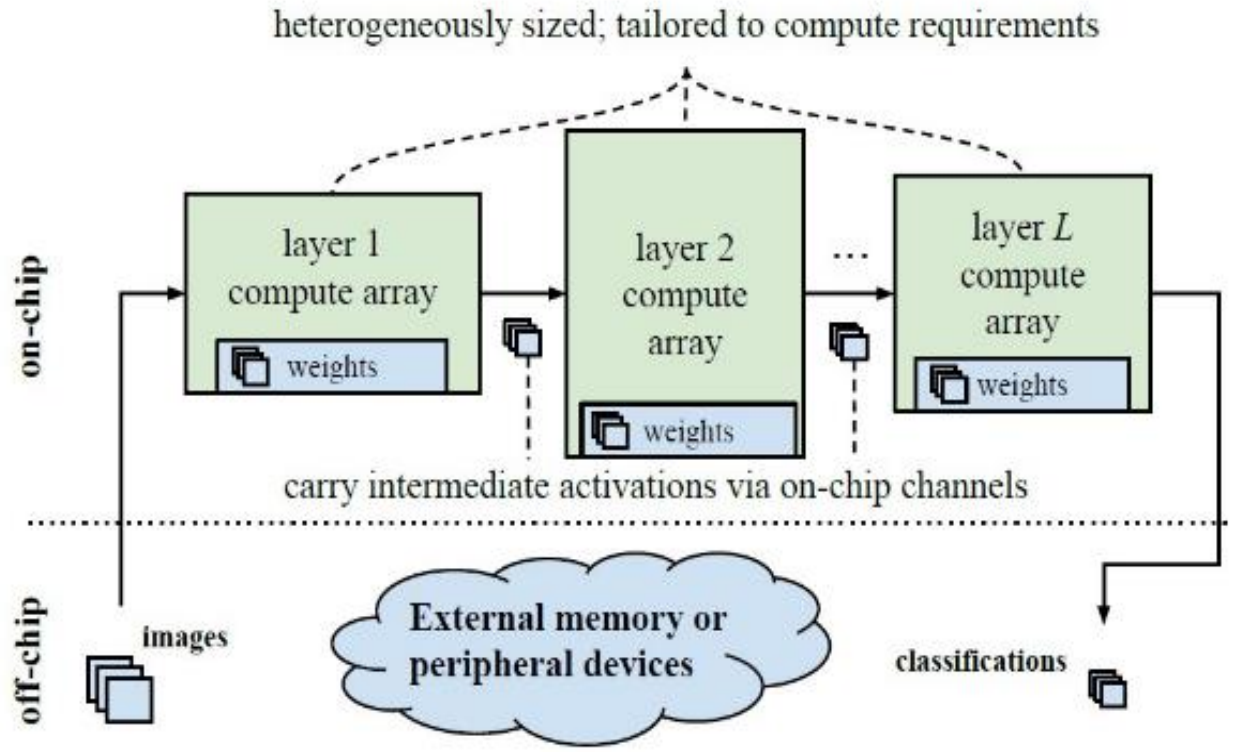
- Determining key NN Accelerator architectural elements

- How much further can we improve NN accelerators with co-design?

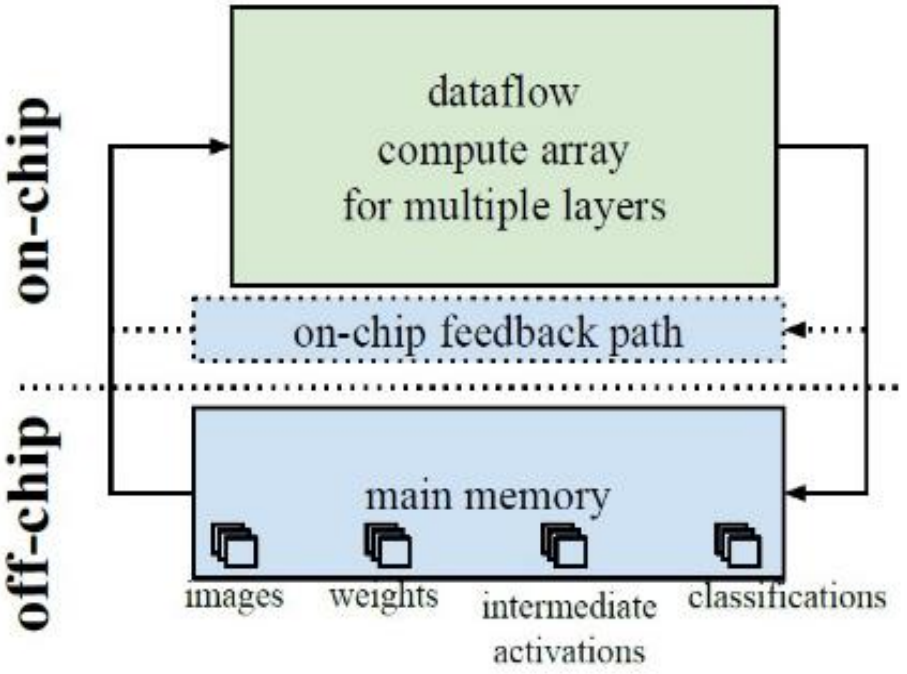
Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Direct mapping of NN to accelerator
  - Layered double-buffer strategy

Dataflow (Spatial) vs Multilayer (Double Buffer) Architecture



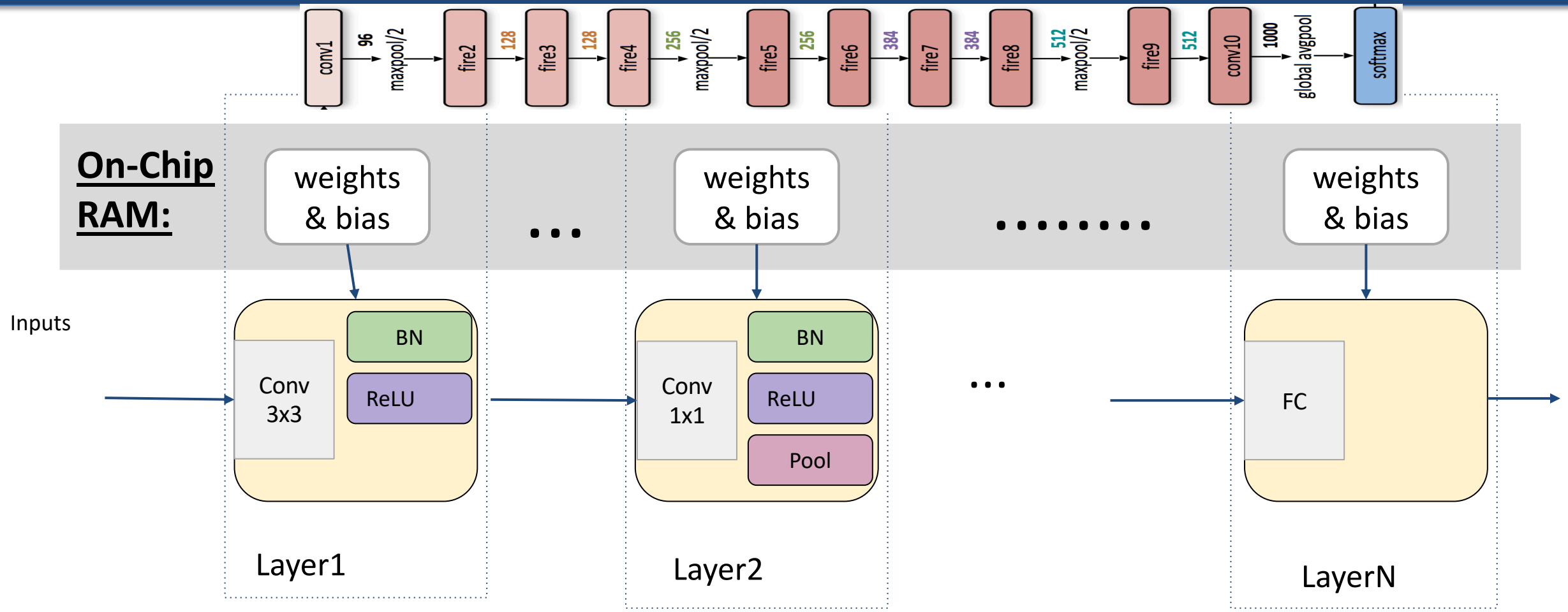
(a) Dataflow Architecture with Per-Layer Tailored Compute Arrays, On-Chip Weights and Activations



(b) Multilayer Offload Architecture with Maximally-Sized Homogeneous Compute Arrays for Different Precisions

FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks, MICHAELA BLOTT et al., ACM Transactions on Reconfigurable Technology and Systems, Vol. 1, No. 1. Publication date: September 2018.

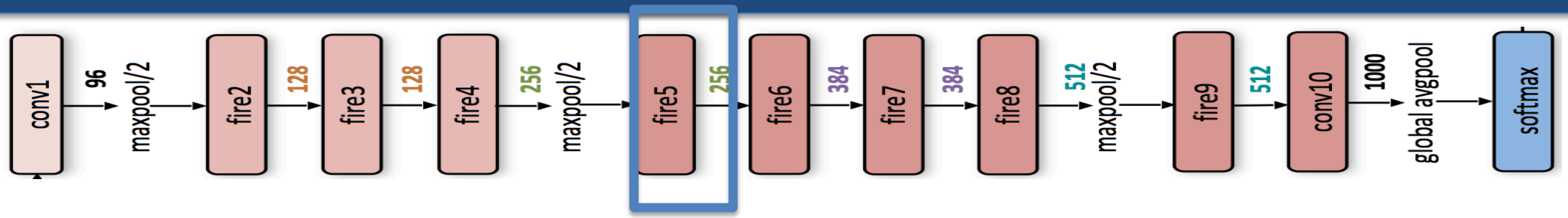
NN to HW Mapping: Spatial mapping
Aka: Dataflow approach



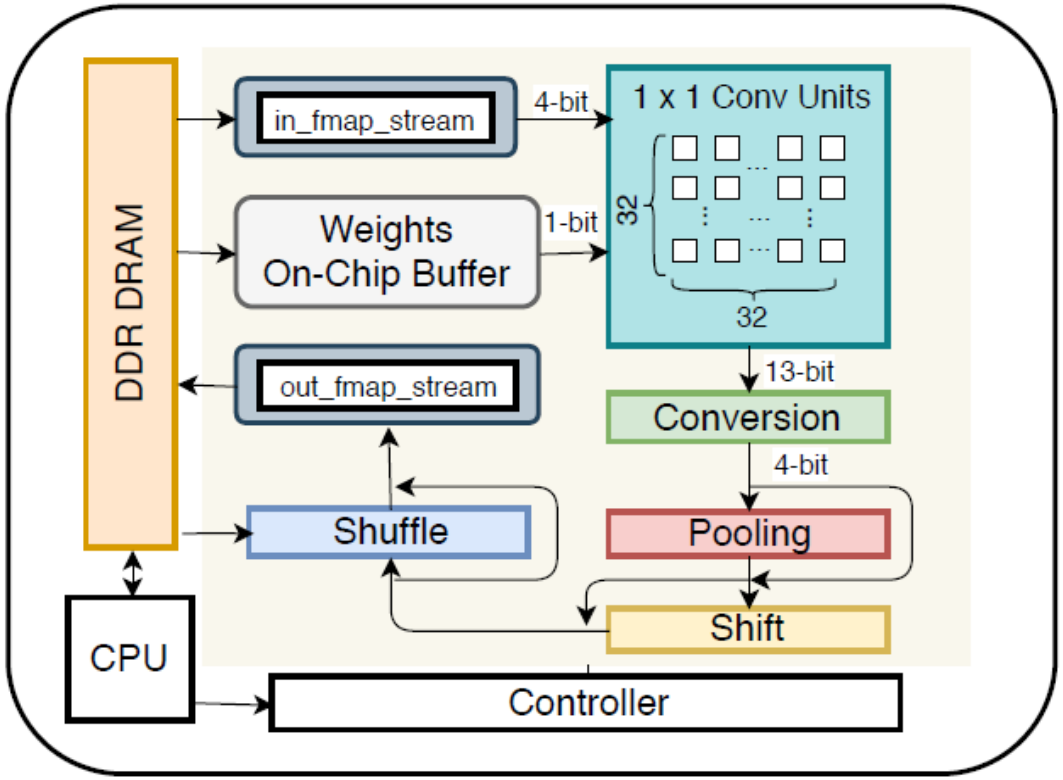
Very efficient if all weights and layers can fit on-chip

Typical usage: small net on FPGA or full-chip NN accelerator

More general mapping strategy: Layer-based/Double-buffered



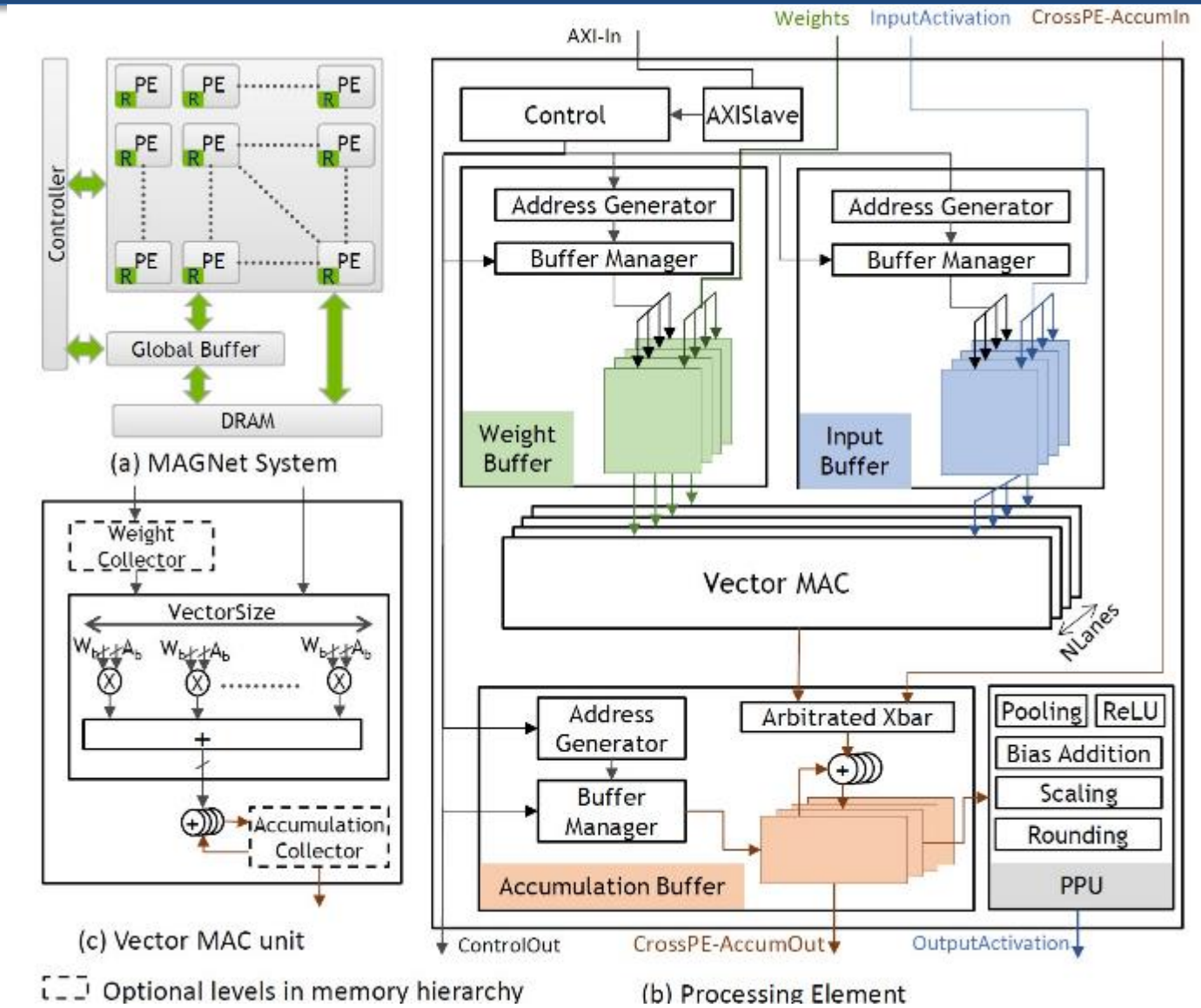
- Process the DNN one layer at a time
- Disadvantages:
  - Must address more challenges to PE—PE or PE– Memory communication
  - Likely to be much more memory traffic overall
- Advantages:
  - accelerator is able to accommodate a very broad range of DNN models



✓Yang, Yifan, Qijing Huang, Bichen Wu, Tianjun Zhang, Liang Ma, Giulio Gambardella, Michaela Blott et al. "Synetgy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas." In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 23-32. ACM, 2019.

MAGNet: Template for a Double-Buffer HW Accelerator

MAGNet: A Modular Accelerator Generator for Neural Networks, Rangharajan Venkatesan† Yakun Sophia Shao† Miaorong Wang‡ Jason Clemons† Steve Dai† Matthew Fojtik† Ben Keller† Alicia Klinefelter† Nathaniel Pinckney† Priyanka Raina⋄ Yanqing Zhang† Brian Zimmer† William J. Dally† Joel Emer†‡ Stephen W. Keckler† Brucek Khailany†, ICCAD '19



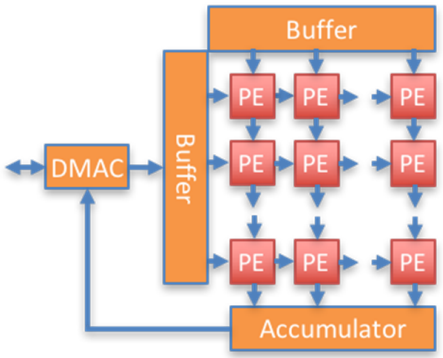
Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function

PE's
number and basic function ...

Feature	Example of trade-offs
Generality	Simple MAC unit to scalar processor, vector units, specialized DSP units
Programmability	Richness and completeness of instruction set, instruction memory size
Organization	Register files: Number – scalar + specialized RFs i.e. weights; Size; Vector unit RF; 2nd level general purpose RF
Local memory, data movement	SRAM data buffer, instruction memory, double buffering to support dataflow, specialized engines to move data between buffers

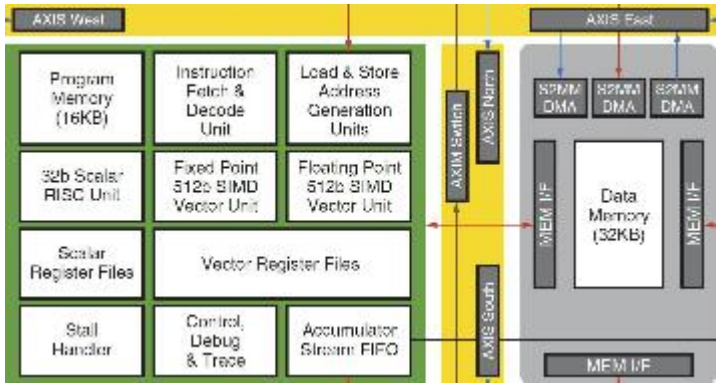
simple ALU/MAC array



Gyrfalcon 2802M -- ~28K PEs

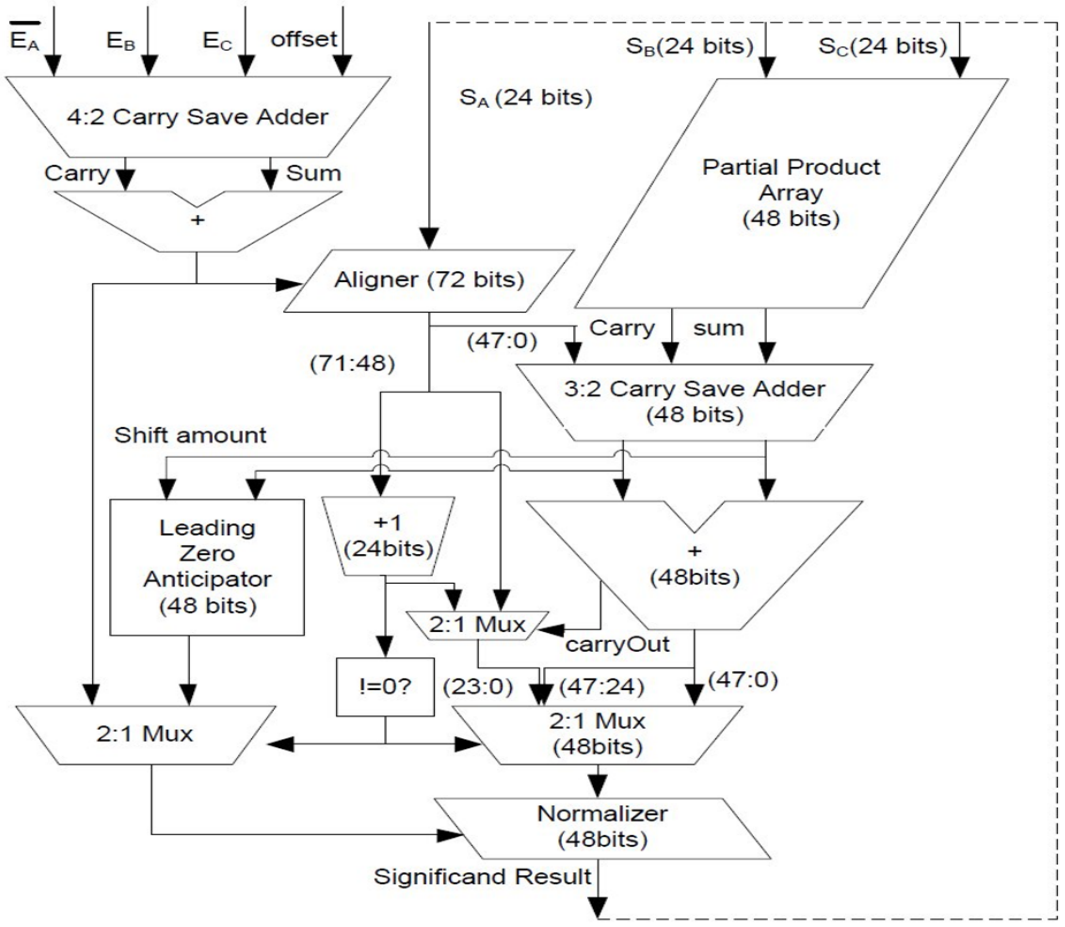
Range of choices
Many intermediate points

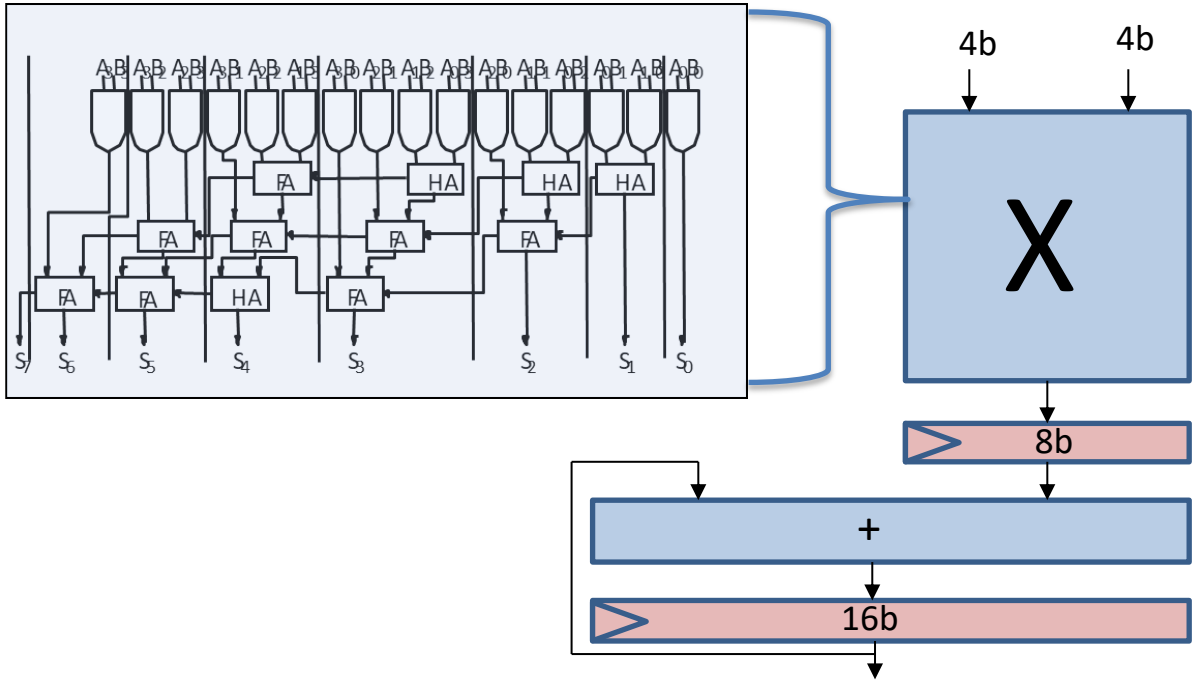
full-featured scalar + vector processor



Xilinx AI Versal VC1092 – 400 PEs
Datacenter/Heavy duty computing

Data types matter a lot!
Data types: 32bit FP MAC vs int4 MAC





- 32 bit FP fused MAC
- 17K gate equivalents

24 X less gates

- int4 MAC
- 4 bit multiply , 16 bit accumulator
- ~700 gates

Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, datatypes, and interconnect

On-chip memory hierarchy & sizing
Keep data (very) close to the processing units

Operation	16 bit (integer)		64 bit (DP-FP)	
	E/op PJ	vs. Add	E/op PJ	vs. Add
ADD	0.18	1.0 ×	5	1.0 ×
Multiply	0.62	3.4 ×	20	4.0 ×
16-Word Register File	0.12	0.7 ×	0.34	0.07 ×
64-Word Register File	0.23	1.3 ×	0.42	0.08 ×
4 K-word SRAM	8	44 ×	26	5.2 ×
32 K-word SRAM	11	61 ×	47	9.4 ×
DRAM	**640**	**3556×**	**2560**	**512 ×**

Ardavan Pedram, Stephen Richardson, Sameh Galal, Shahar Kvatinsky, and Mark A. Horowitz, "Dark Memory and Accelerator-Rich System Optimization in the Dark Silicon Era," IEEE Design and Test Magazine Special Issue on Dark Silicon, April 2017.
https://arxiv.org/pdf/1602.04183.pdf

Memory (Bandwidth) Wall

Memory Location	Size	Relative (Absolute) Bandwidth
L0 – Memory next to Processing Elements, Distributed on-chip	KB to 10s KB/PE	1x (2000 TB/s)
L1 – Share buffers, on-chip	100 KBs to 1 MB	1/10 (200 TB/s)
L2 – Global buffer, on-chip	10s MB	1/100 (20 TB/s)
HBM2 DRAM, off-chip, in-package	10s GB	1/2000 (1 TB/s)

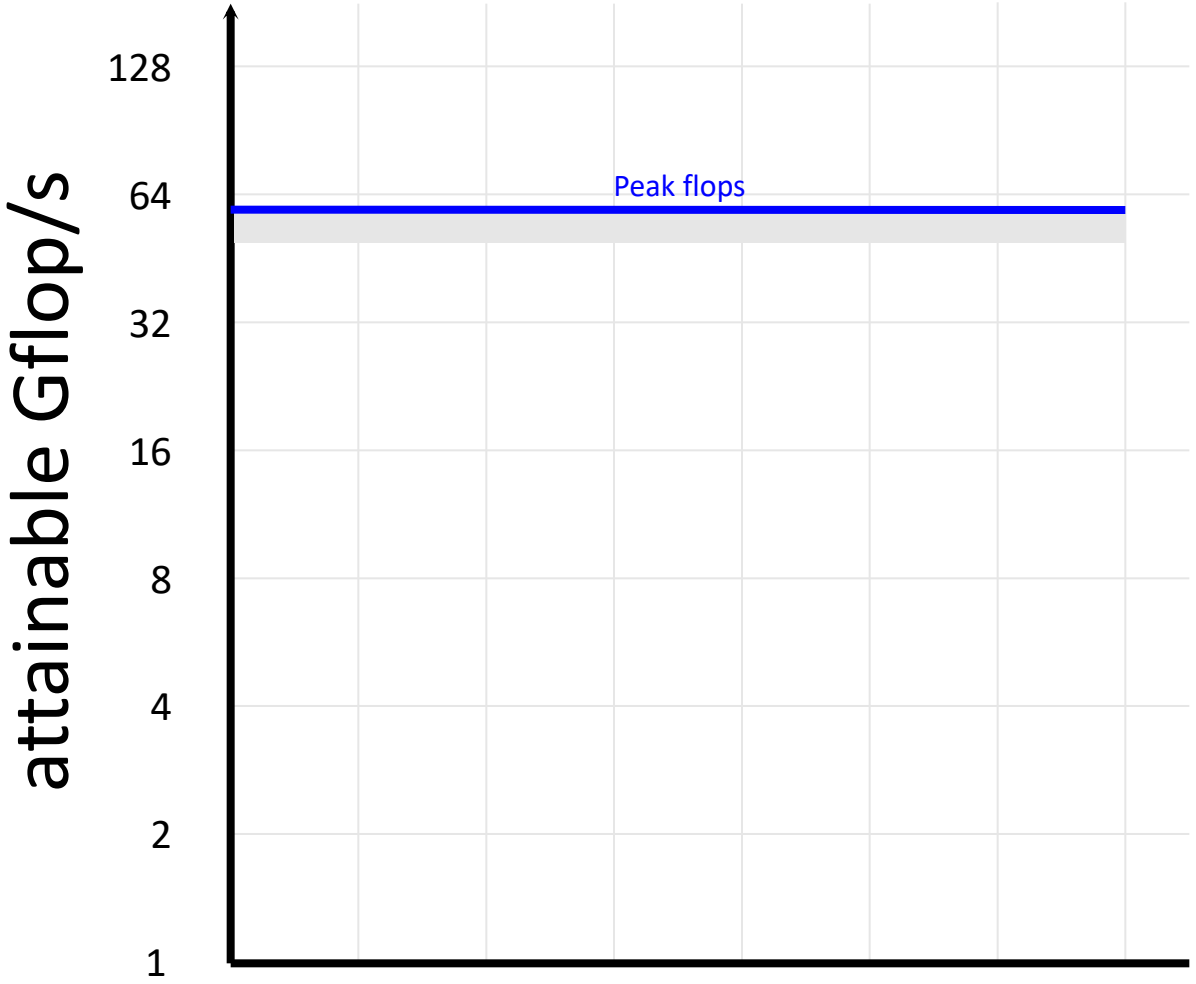
Adapted from “DaVinci: A Scalable Architecture for Neural Network Computing”, Heng Liao, Jiajin Tu, Jing Xia, Xiping Zhou, Huawei, Hot Chips 2019

Williams Roofline Model

- For decades programmers and architects did back-of-the-envelope calculations on compute vs communication at various levels of the memory hierarchy
  - Processor to register file
  - On-chip L1, L2 (L3?) caches
  - Off-chip DRAM
  - Interprocessor communication
- UC Berkeley grad student Sam Williams gave a simple model, known as the Roofline Model, for reasoning about these issues
  - Aimed at reasoning about kernel arithmetic intensity and interprocessor communication, we can use it across the memory hierarchy

Williams, Samuel, Andrew Waterman, and David Patterson. *Roofline: An insightful visual performance model for floating-point programs and multicore architectures*. No. LBNL-2141E. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009.

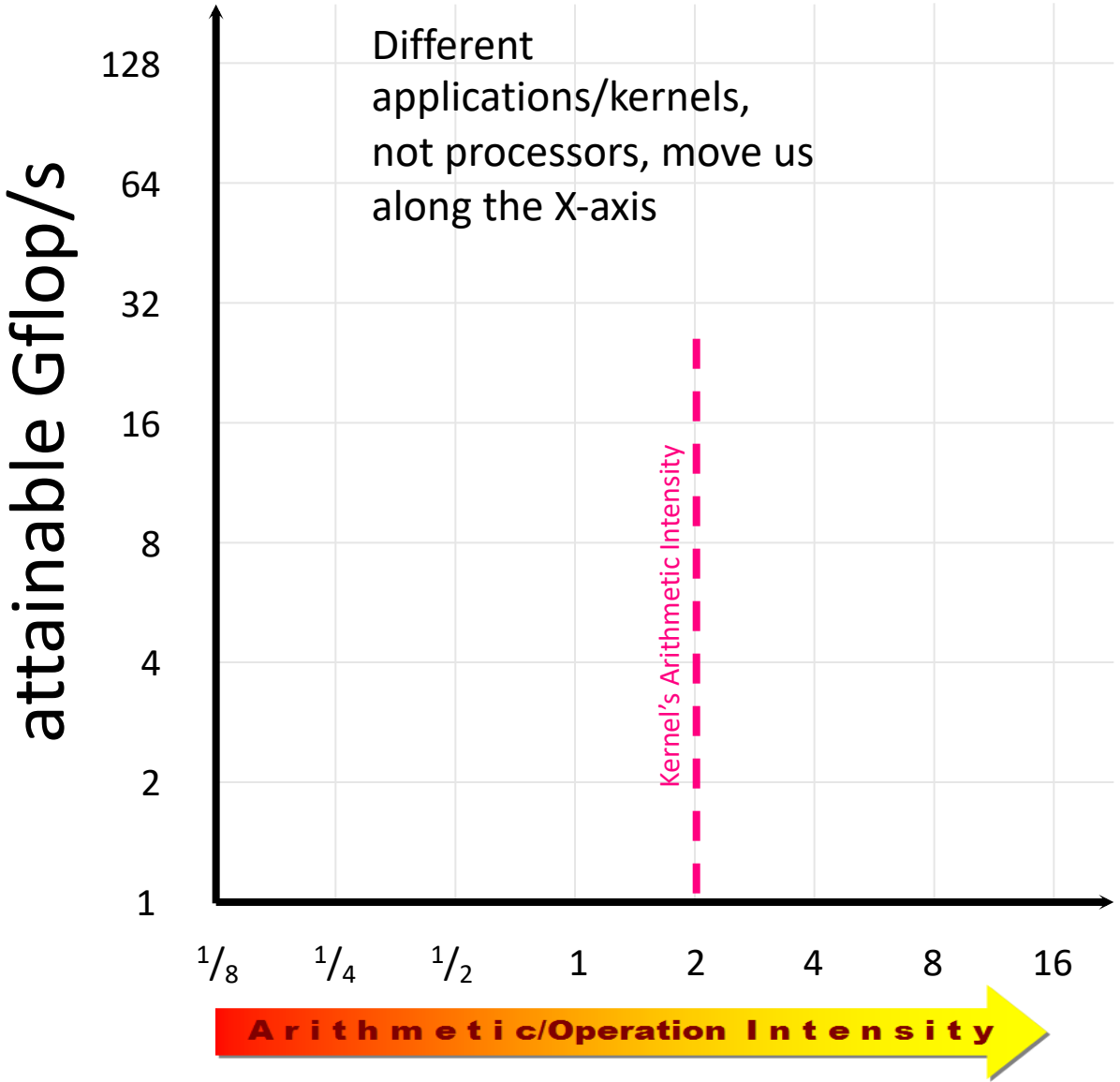
Roofline Model: y-axis



- The y-axis describes the attained performance
- It's easy to add the "peak performance" as an upper bound

Williams, Samuel, Andrew Waterman, and David Patterson. *Roofline: An insightful visual performance model for floating-point programs and multicore architectures*. No. LBNL-2141E. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States), 2009.

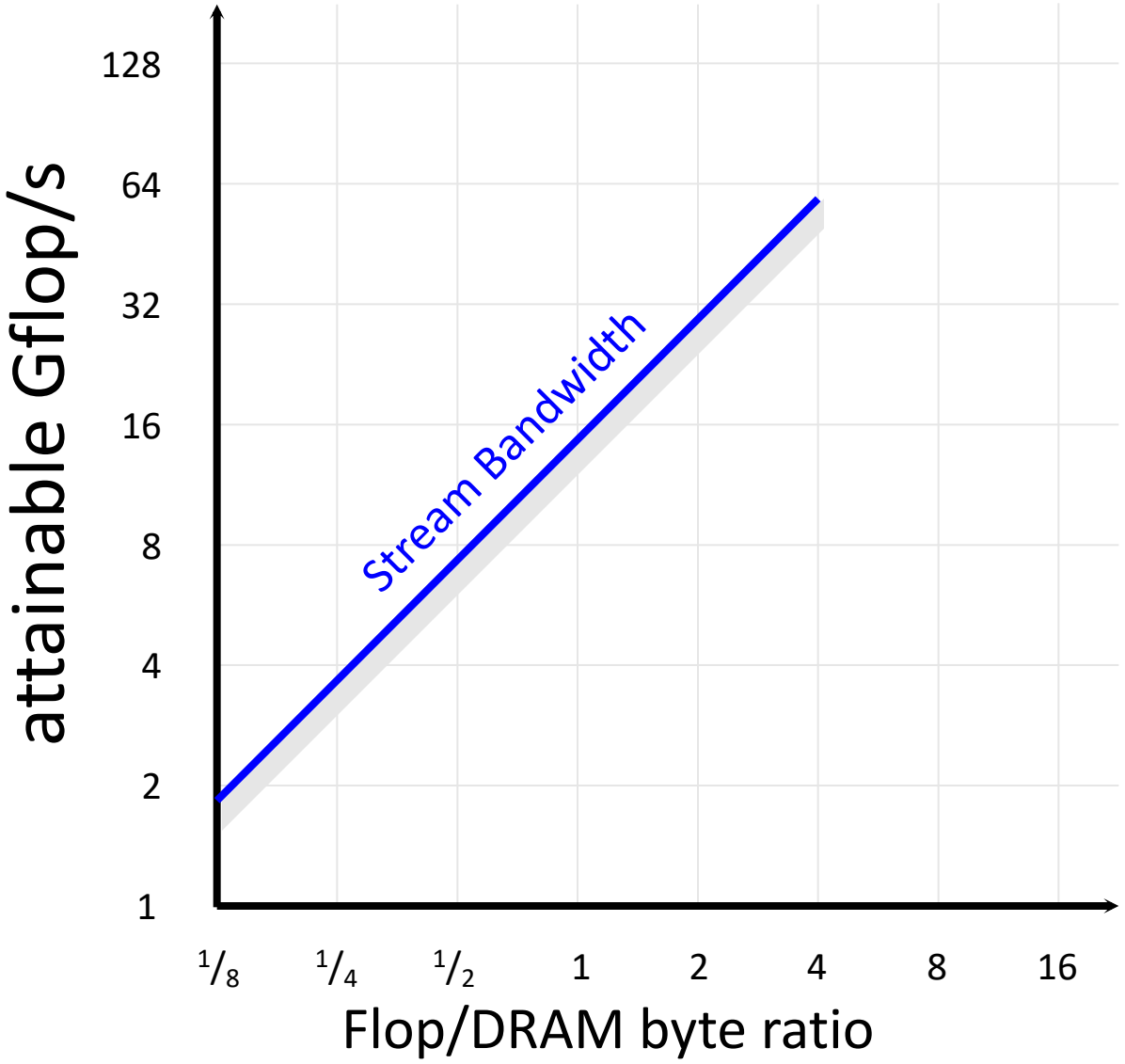
Roofline Model: x-axis



- The x-axis tells indicates for this particular application/kernel, for each floating-point operation (flop), how many bytes (B) must be fetched
- For example if we have to fetch two double precision floating-point numbers for each floating point operation, then:
- 1 double precision float: 8 bytes (64 bits)
- 2 double precision float: 16 bytes
- 1 flop requires 16 DRAM bytes

Flops/byte

Bandwidth as Slope



Bandwidth is represented as a slope of

Peak Flops/ AI

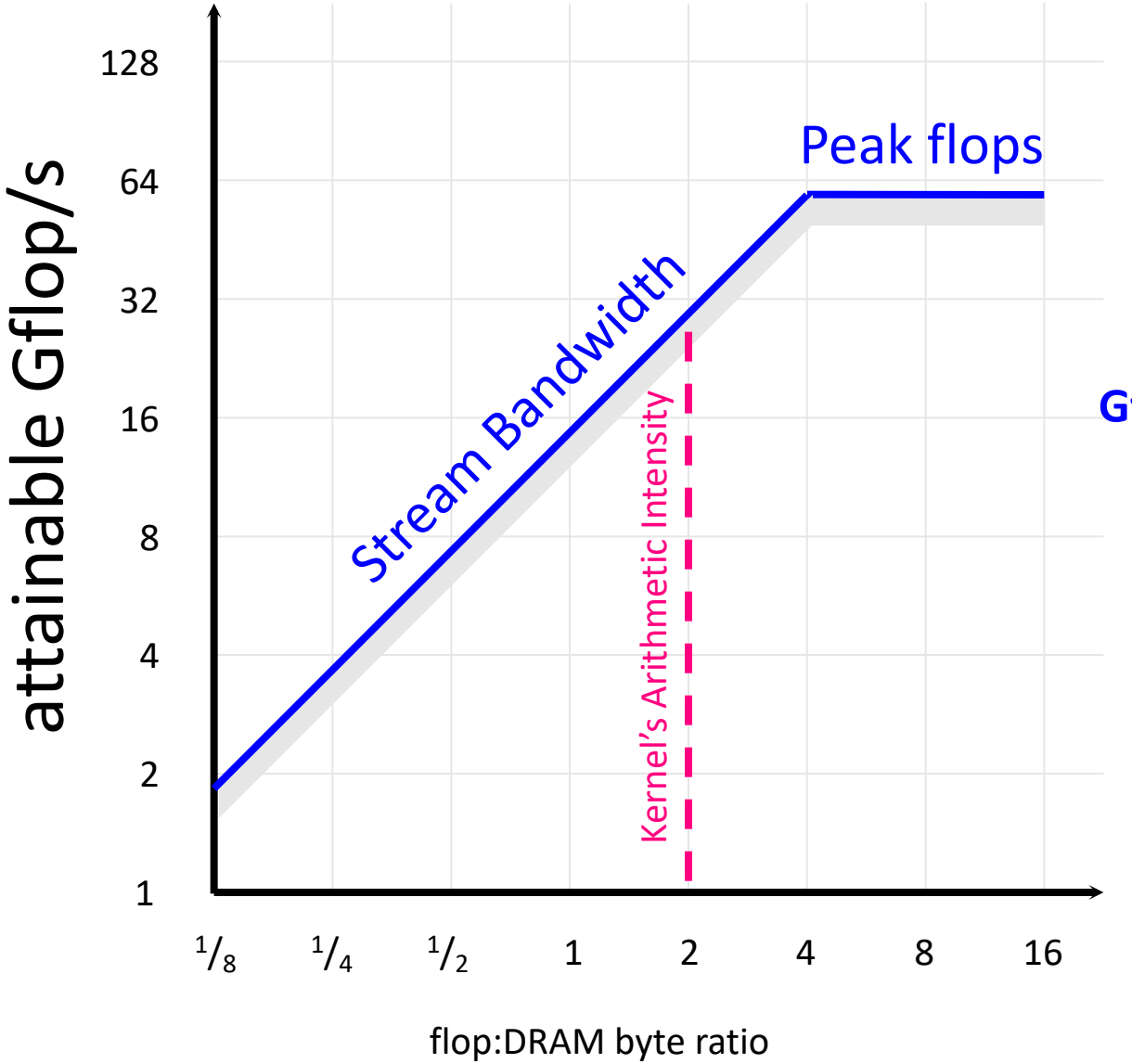
It is a given by the system configuration/architecture

- Remember $m = \frac{y}{x}$

Example

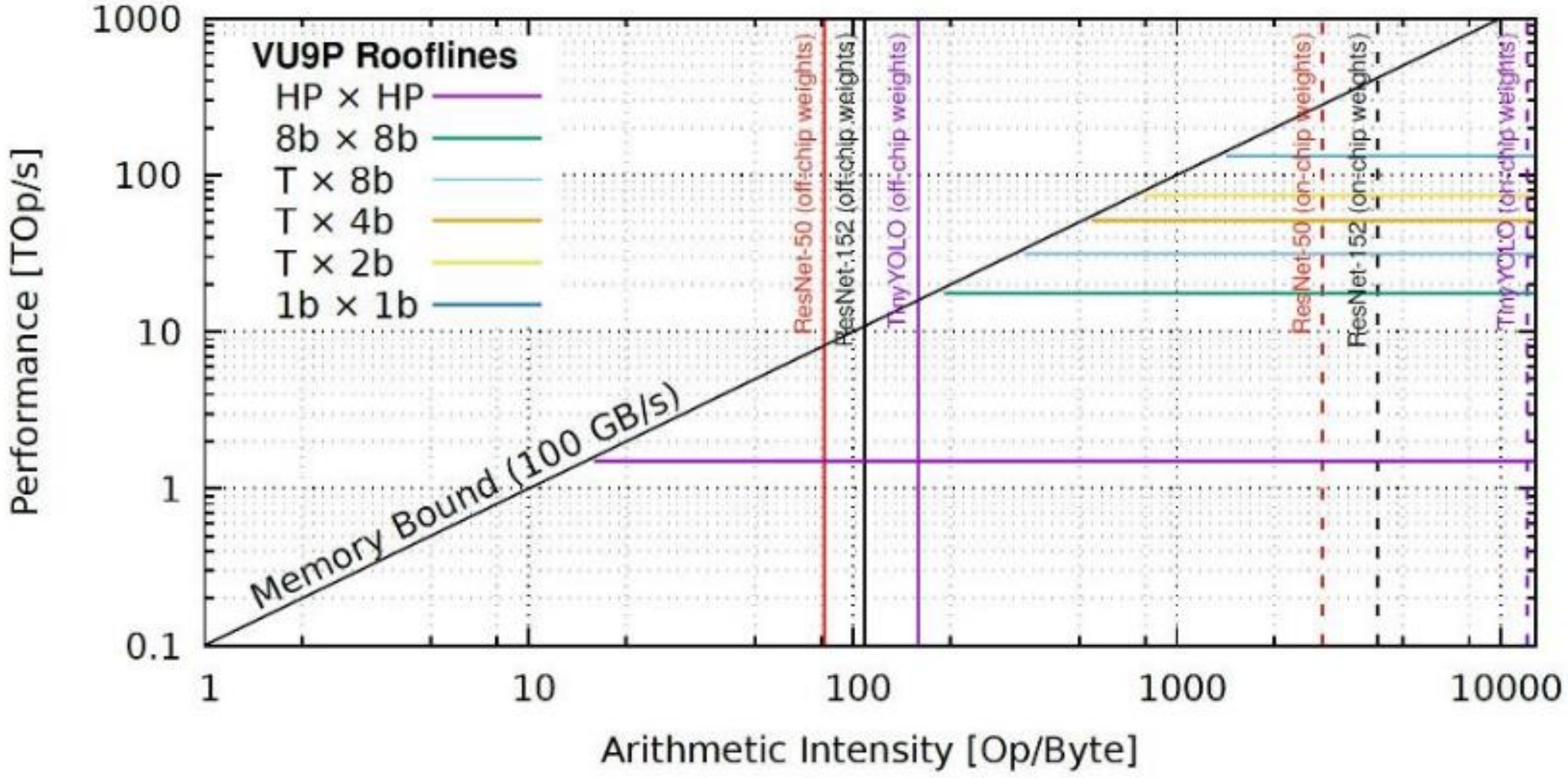
- m = y/x = 16 Gbytes/s
- y= $\frac{2G\ Flop}{second}$
- x = $\frac{1\ Flop}{8\ bytes}$

Bandwidth Meets Arithmetic Intensity



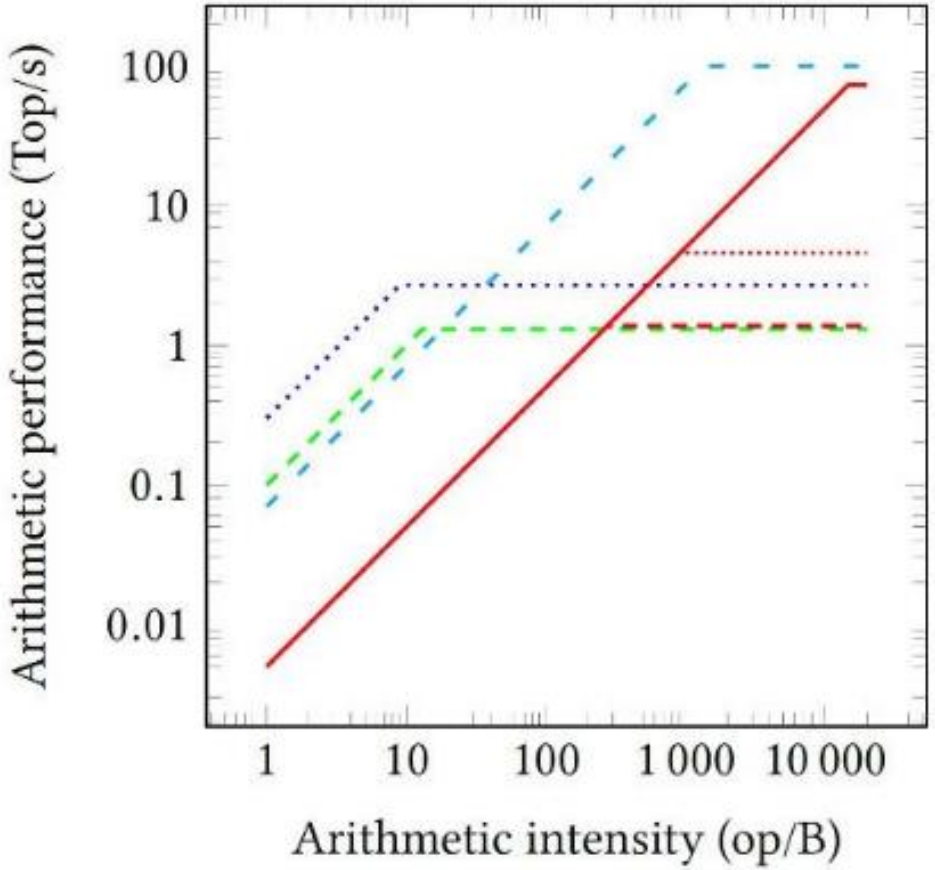
Gflop/s(AI) = min { Peak Gflop/s, StreamBW * AI

Roofline Model vs. Data type Vs Different Models (Blott)

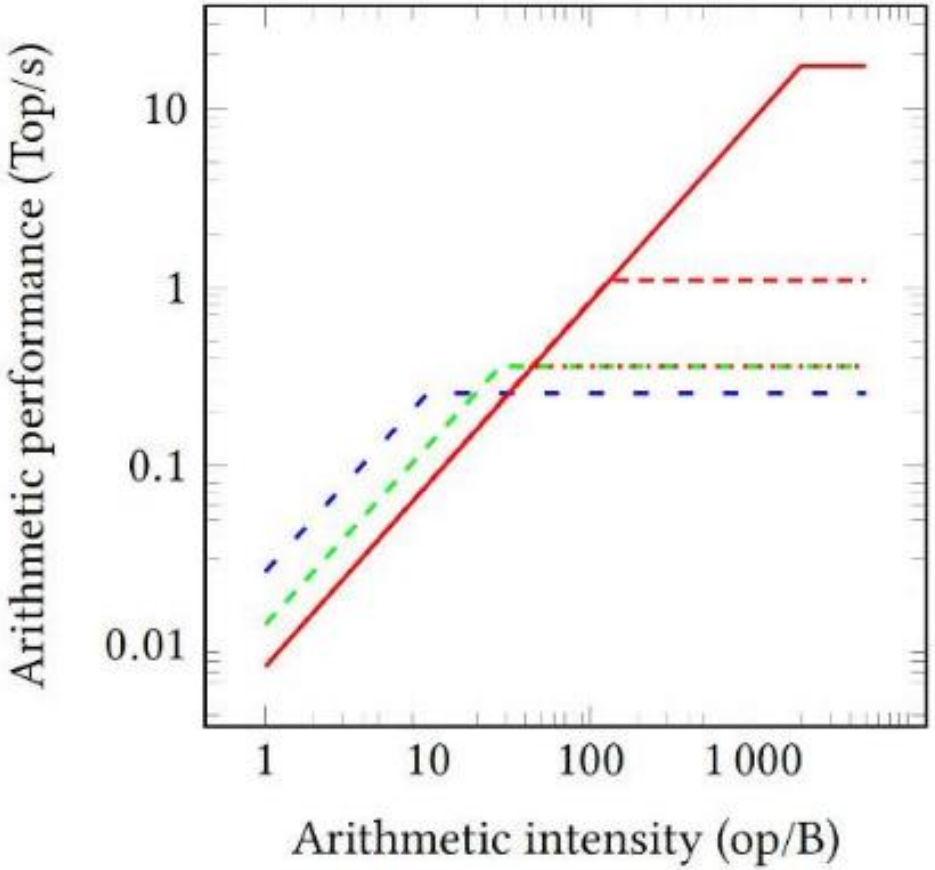


FINN-R: An End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks, MICHAELA BLOTT et al., ACM Transactions on Reconfigurable Technology and Systems, Vol. 1, No. 1. Publication date: September 2018.

Roofline Model vs. Data Types



(a) Datacentre-scale platforms: 18-core Intel Haswell CPU (- - -), Nvidia Tesla K80 GPU (·····), Google TPU ASIC (– –) and Xilinx Kintex UltraScale KU115 FPGA with 16-bit (– – –), eight-bit (·····) and one-bit (——) fixed-point weights [66, 141].



(b) Embedded-scale platforms: Nvidia Jetson TX1 GPU (– –), TI Keystone II DSP (- - -) and Xilinx Zynq ZC706 FPGA with 16-bit (·····), eight-bit (– – –) and one-bit (——) fixed-point weights [58].

Deep Neural Network Approximation for Custom Hardware: Where We've Been, Where We're Going, ERWEI WANG et al., ACM Computing Surveys, Vol. 1, No. 1, Article 1. Publication date: January 2019.

Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy,

On-chip memory hierarchy & sizing: Consider Small Register Files for Higher Energy Efficiency

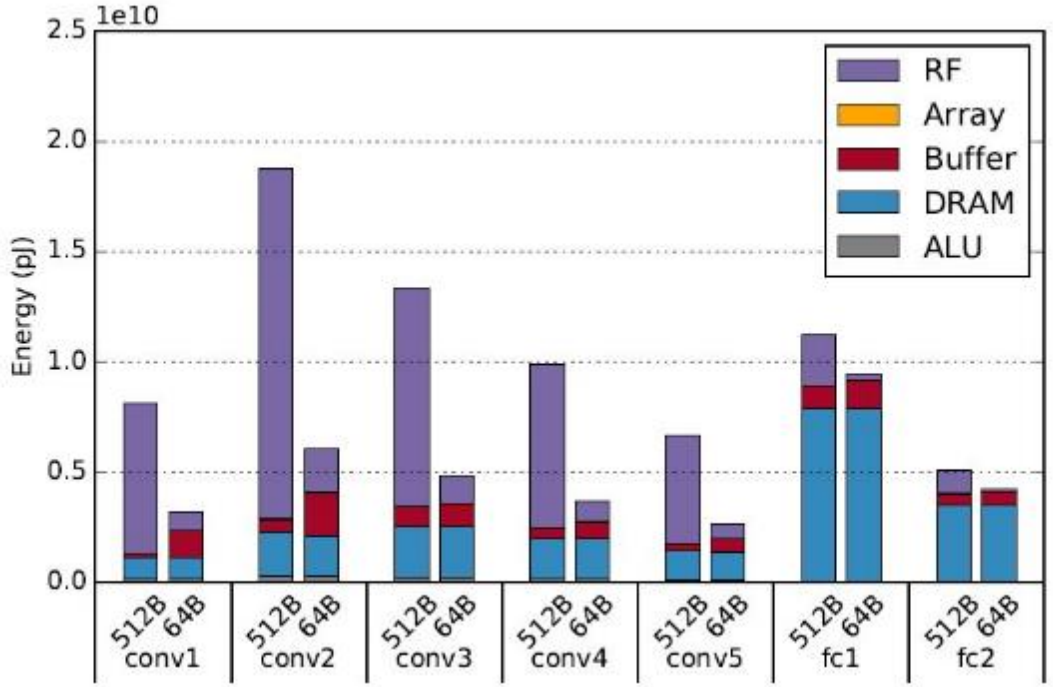
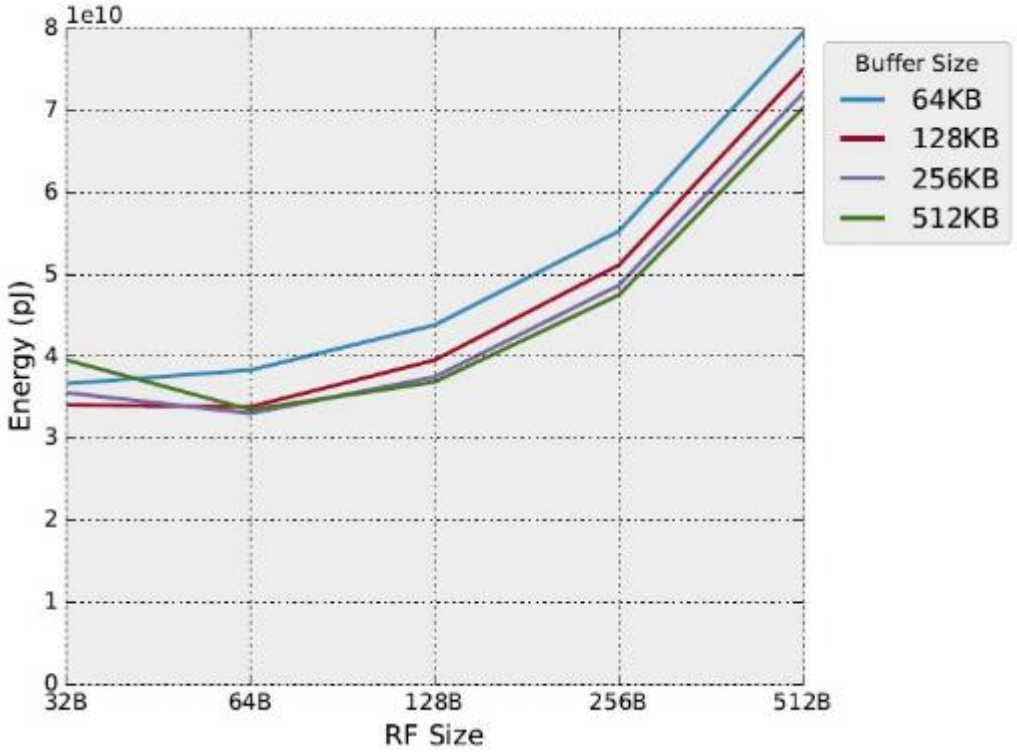


Figure 12
Xuan Yang, Mingyu Gao, Jing Pu, Ankita Naya, Qiaoyi Liu, Steven Emberton Bell, Jeff Ou Setter, Kaidi Cao, Heonjae Ha, Christos Kozyrakis and Mark Horowitz, "DNN Dataflow Choice Is Overrated", arXiv:1809.04070v1. [Yang-Horowitz 2018]

- **Consider smaller Register Files: 64B is ~2.6x more energy efficient overall.**
- Smaller RF has much lower energy cost per access, yielding big savings for Conv layers.
- More RF "misses", but these go to large on-chip global buffer.
- As a result, DRAM access cost does not change.

On-chip memory hierarchy & sizing: Consider Large Global Buffer



- **Global buffer catches "misses" from Register File, minimizing expensive DRAM accesses.**
- Major savings going from 64kB to 128kB, some gain up to 256kB.
- No gains once DRAM accesses are minimized – input data read once, output data written once.

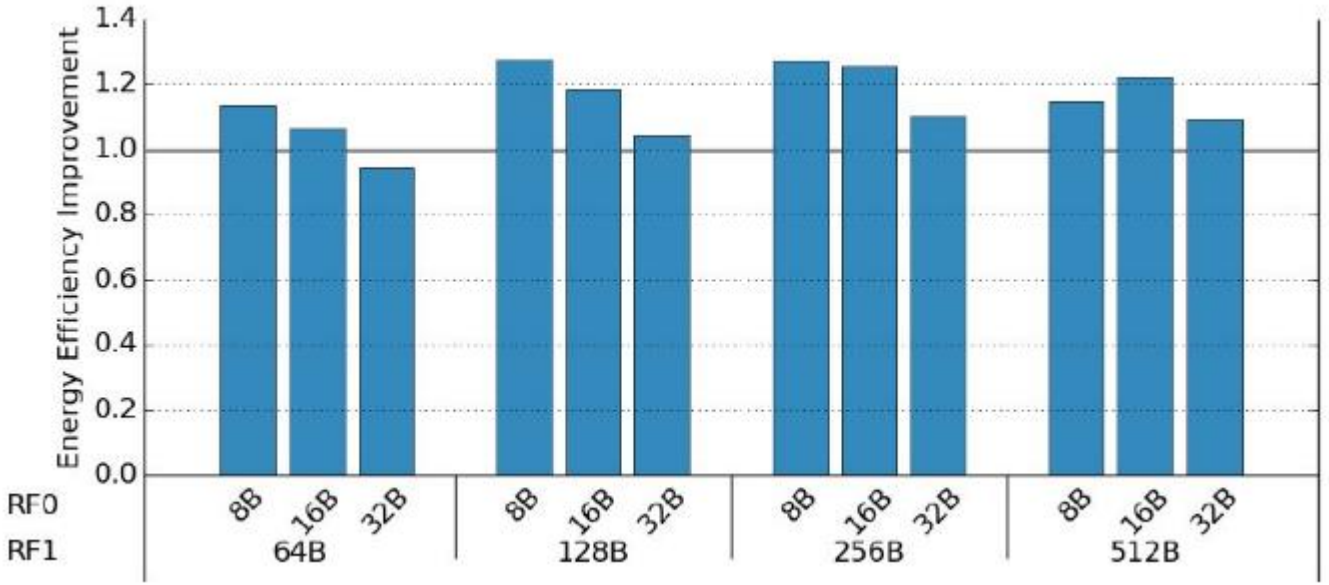
Xuan Yang, Mingyu Gao, Jing Pu, Ankita Naya, Qiaoyi Liu, Steven Emberton Bell, Jeff Ou Setter, Kaidi Cao, Heonjae Ha, Christos Kozyrakis and Mark Horowitz, "DNN Dataflow Choice Is Overrated", arXiv:1809.04070v1 [Yang-Horowitz 2018]

GPU "Inverted" Memory Hierarchy

Product	K40	M40	Tesla P100	Tesla V100	T4
GPU	GK180 Kepler	GM200 Maxwell	GP100 Pascal	GV100 Volta	TU104
Year	2013	2015	2016	Early 2018	Late 2018
Streaming Multiprocessors	15	24	56	80	40
Shared memory/SM (L1 cache)	16kB-48kB	96kB	64kB	Up to 96kB	96kB
RegFile/SM (per GPU)	256kB (3840kB)	256kB (6144kB)	256kB	256kB	256kB
L2 Cache	1536kB	3072kB	4096kB	6144kB	4096kB
Shared Memory per SM : RF per SM	0.18x	0.37x	0.25x	0.37x	0.37x
L2 cache : Shared Memory per SM	32x	48x	64x	64x	42x
L2 cache : RF per SM	6x	12x	16x	24x	16x

- ***Very different from Horowitz recommendation; RF very large; Shared memory typically smaller than RF!***

On-chip memory hierarchy & sizing: Consider Adding an Extra Level of Memory Hierarchy



Xuan Yang, Mingyu Gao, Jing Pu, Ankita Naya, Qiaoyi Liu, Steven Emberton Bell, Jeff Ou Setter, Kaidi Cao, Heonjae Ha, Christos Kozyrakis and Mark Horowitz, "DNN Dataflow Choice Is Overrated", arXiv:1809.04070v1 [Yang-Horowitz 2018]

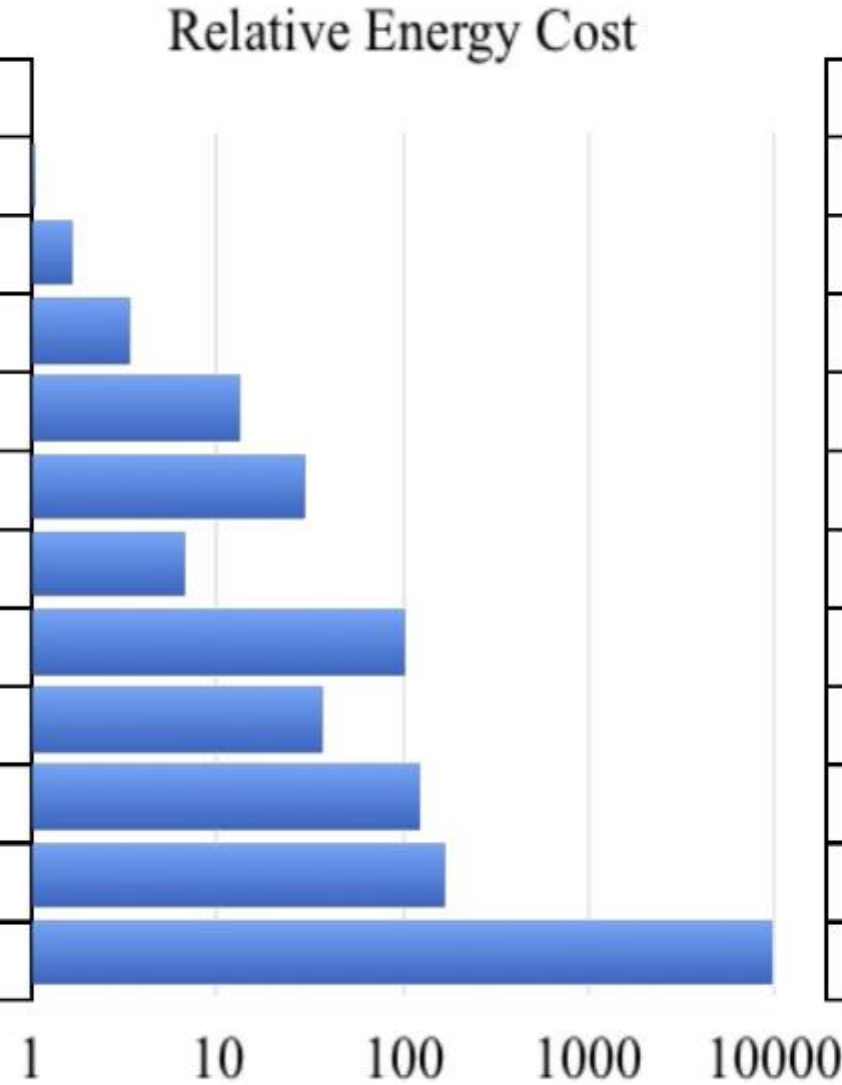
- **Adding a 2nd level RegFile, private to each PE, saves ~25% energy overall.**
- Savings are most pronounced for Conv layers, about 30%.
- Ratio of memory sizes between adjacent memory levels should be at least 8x-16x, i.e. 16B/256B.
- DNNs can support larger ratios than CPU caches thanks to predictable data patterns and perfect prefetching.

Key Architectural Decisions

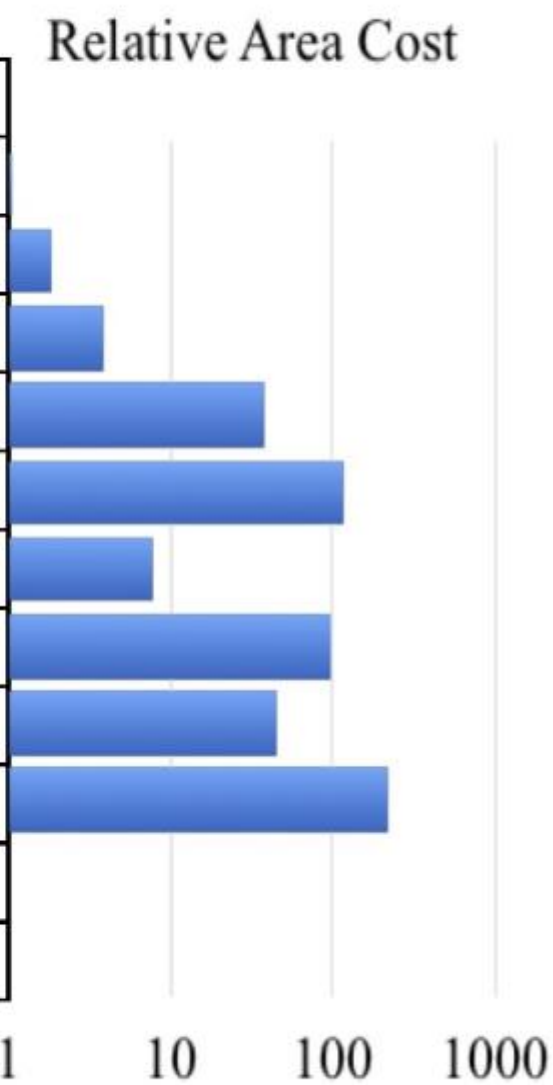
- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, **datatypes**

Cost of Operations

Operation:	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Mult	0.2
32b Mult	3.1
16b FP Mult	1.1
32b FP Mult	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640



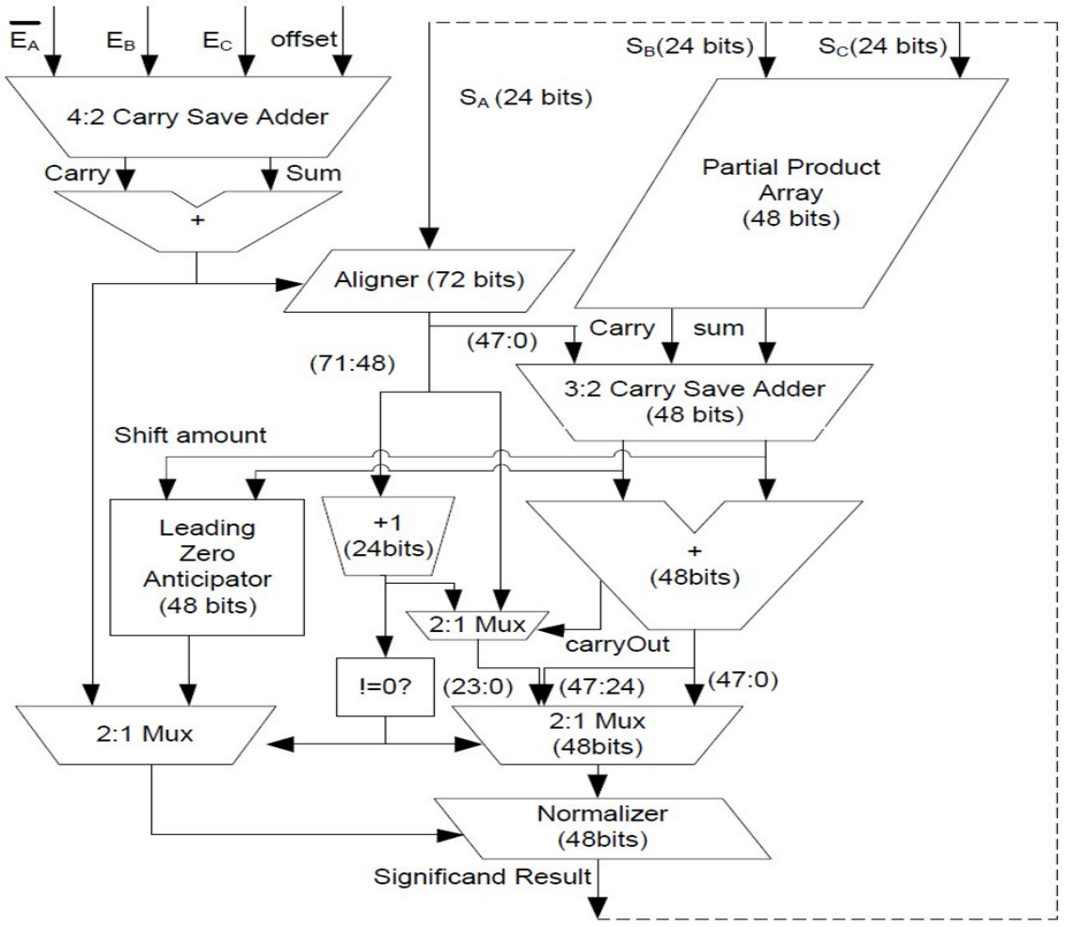
Area (μm^2)
36
67
137
1360
4184
282
3495
1640
7700
N/A
N/A

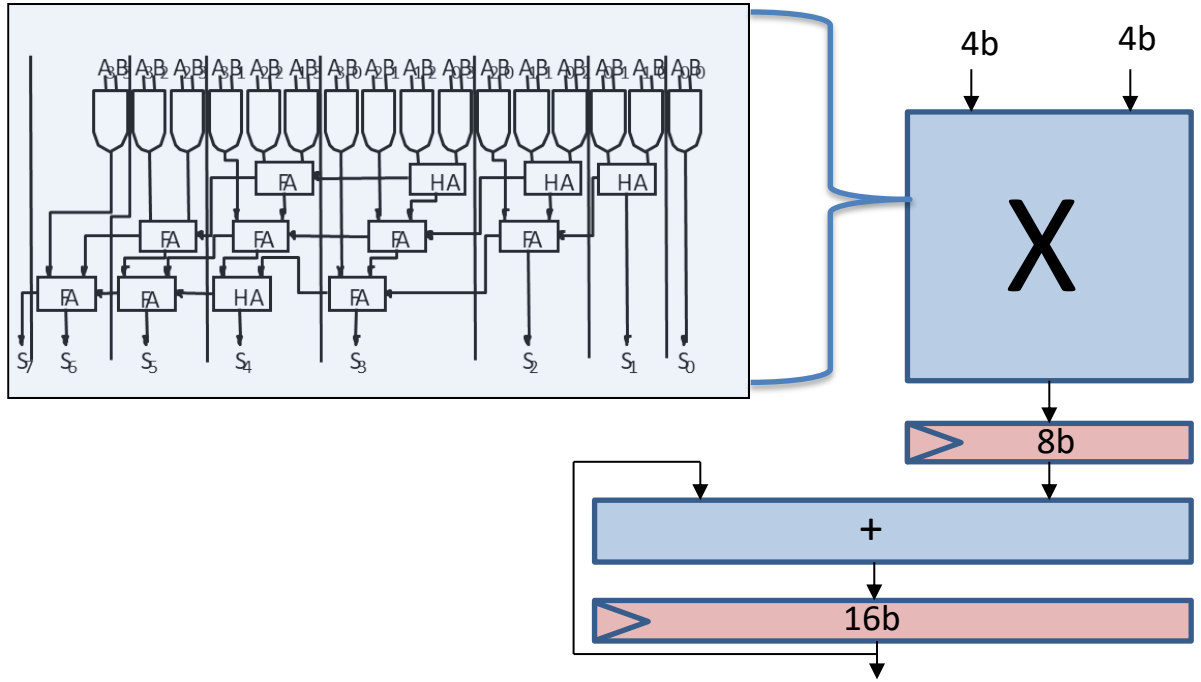


Energy numbers are from Mark Horowitz "Computing's Energy Problem (and what we can do about it)", ISSCC 2014

Area numbers are from synthesized result using Design Compiler under TSMC 45nm tech node. FP units used DesignWare Library.

Data types: 32bit FP MAC vs int4 MAC



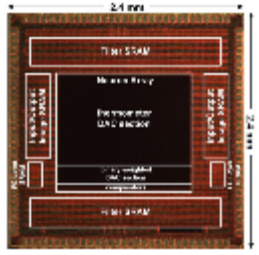
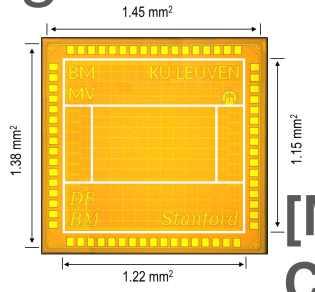
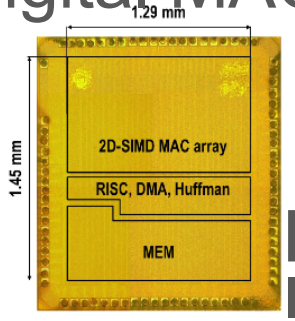


- 32 bit FP fused MAC
- 17K gate equivalents

24 X less gates

- int4 MAC
- 4 bit multiply , 16 bit accumulator
- ~700 gates

Circuit level implications [Marian Verhelst, KU Leuven]

	analog MAC		or 1-bit digital MAC		or multi-precision digital MAC
		*With Stanford (Murmann)*			
	[Bankman, ISSCC18]		**[Moons, CICC18]**		**[Moons, ISSCC17]**
Area	large		small		medium
Energy	500TOPS/W		200TOPs/W		16b: 0.5 TOPs/W 8b: 1 TOPs/W 4b: 5 TOPs/W 2b: 10 TOPs/W
Flexibility	low		medium		high
Accuracy					

20 – 400 X

50 – 1000 X

KK: Low hanging fruit is in memory/arithmetic support for low bit-precisions

Resulting Datatypes and Memory Hierarchy for NN Accelerator (Inference)

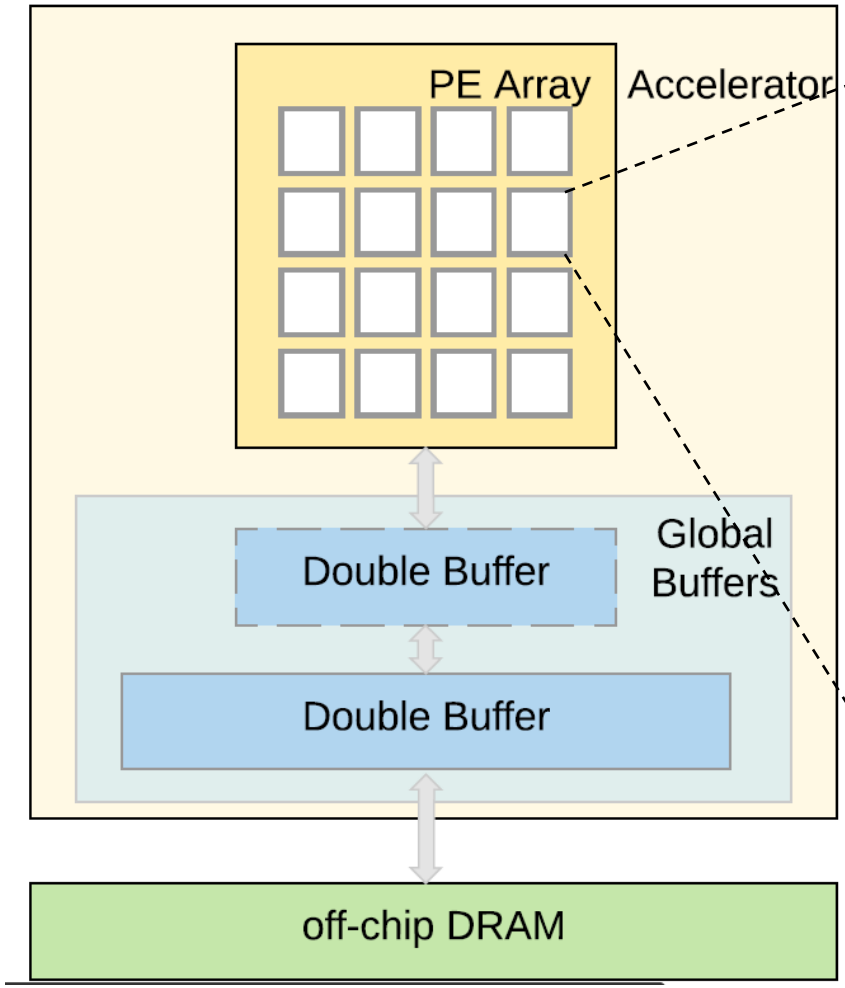
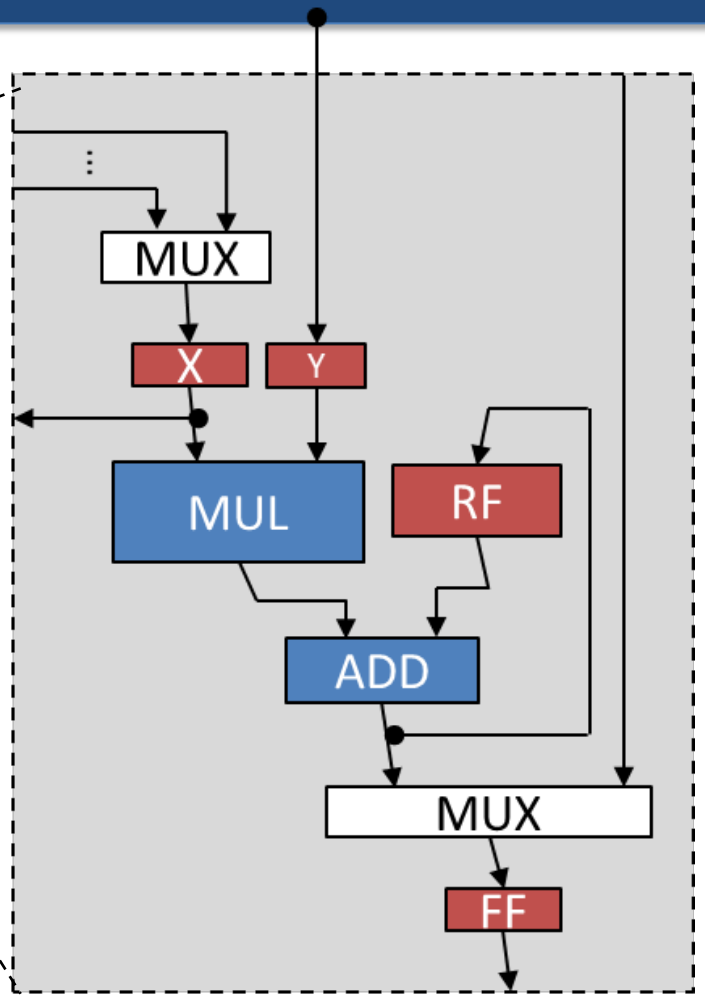


Figure 3, [Yang-Horowitz 2018]



one PE
Squeezelerator,
DAC 2018

KK: Data Types

- 4, 8, 16 bits
- 32 bits?
- 1,2 bits?

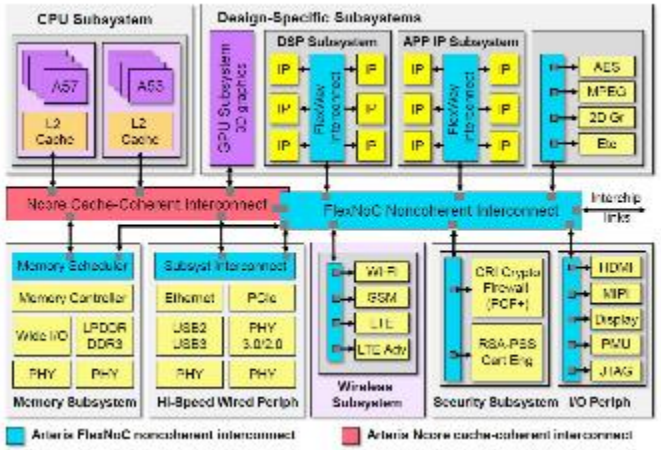
Reg. Size	Energy (pJ)
16 B	0.03
32 B	0.06
64 B	0.12
128 B	0.24
256 B	0.48
512 B	0.96

Table IV
[Yang-Horowitz 2018]

Key Architectural Decisions

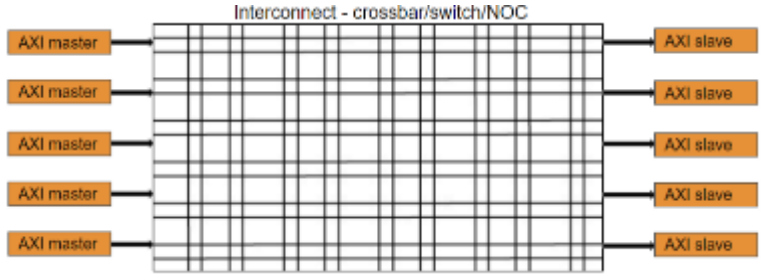
- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, datatypes, and interconnect

On-Chip Interconnect?

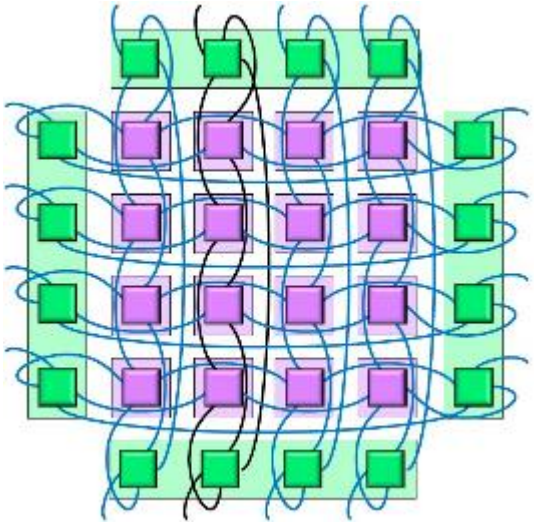


Microprocessor Report, April 2017

Traditional NoCs scale poorly to many-core NN accelerators…..



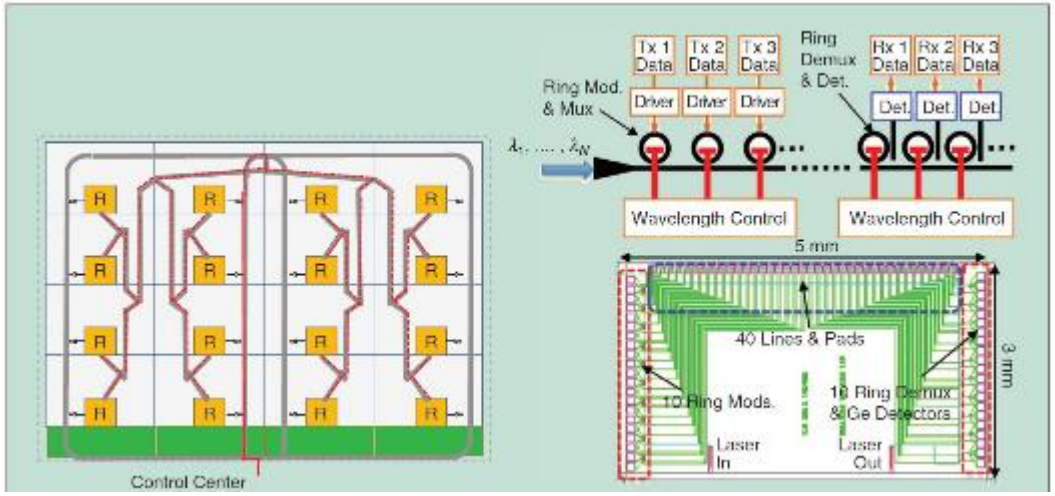
https://anysilicon.com/understanding-amba-bus-architechture-protocols/



[Kalray Mesh] Microprocessor Report, Feb. 2015

…New approaches are needed to support 1,000s of cores.

Multicast
Novel Architectures
Photonics?
On-chip Waveguides?



[Optical NoC] A Survey of Emerging Interconnects for On-Chip Efficient Multicast and Broadcast in Many-Cores, IEEE Circuits and Systems Magazine, Q1 2016.

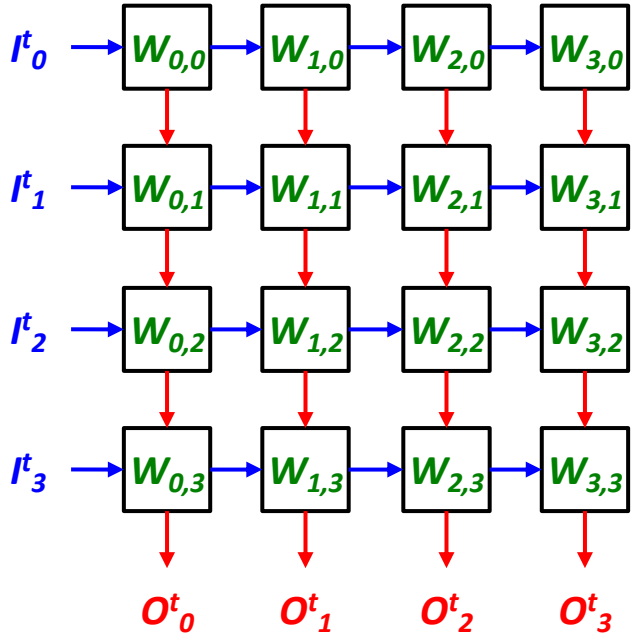
Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, datatypes, and interconnect
  - Economical sizes of register files; sufficient memory size and hierarchy
- PE—PE — Memory dataflow

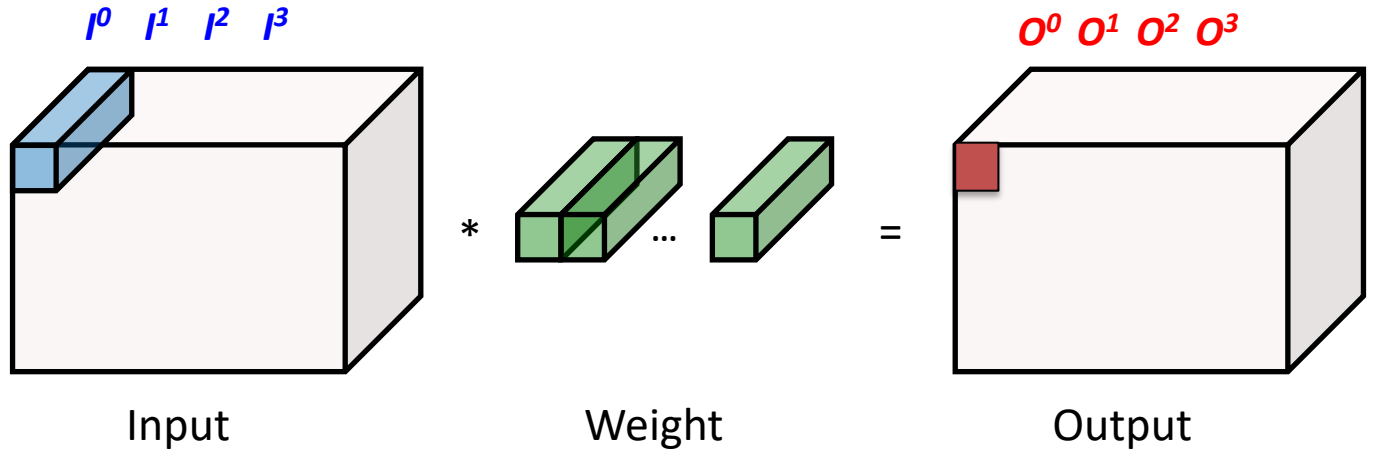
Systolic Array Weight Stationary Example: TPU [1] (Google)

- "Matrix Multiply Unit" performs general matrix-vector multiplications.
- The weight matrix is preloaded in the PE Array.
- A stream of input activation vectors is passed to each column of the array.
- Partial sums of PEs are vertically propagated

Output = Weight-Matrix x Inputs



$$\begin{bmatrix} O_0^t \\ \vdots \\ O_{k-1}^t \end{bmatrix} = \begin{bmatrix} W_{0,0} & \cdots & W_{0,c-1} \\ \vdots & \ddots & \vdots \\ W_{k-1,0} & \cdots & W_{k-1,c-1} \end{bmatrix} \begin{bmatrix} I_0^t \\ \vdots \\ I_{c-1}^t \end{bmatrix}$$

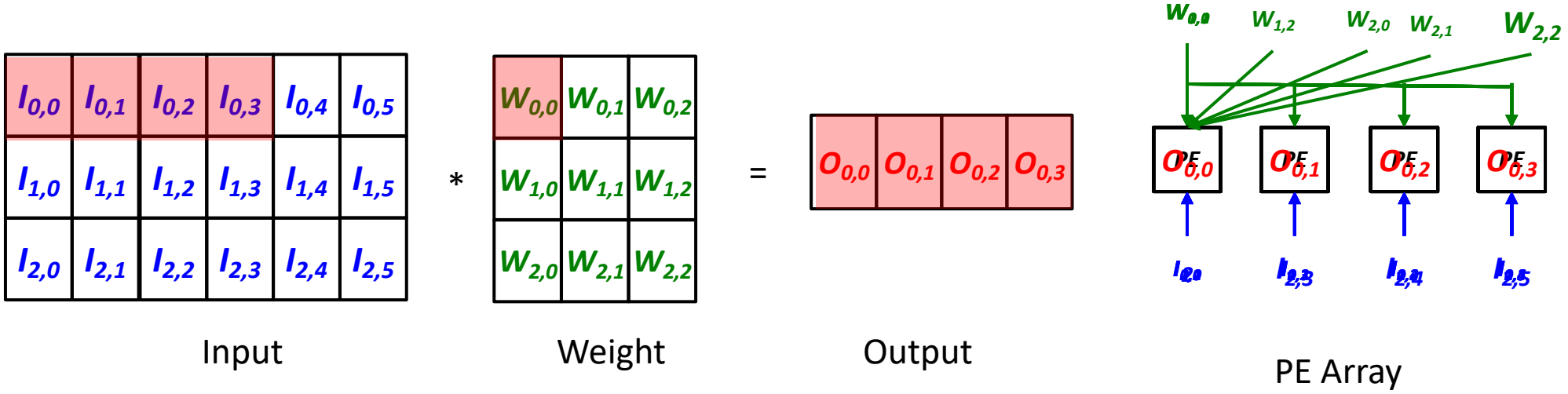


[1] N. Jouppi, et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," 2017.

Systolic Array Output Stationary Example: ShiDianNao [1] (Cambrian)

- Each PE in "Neural Functional Unit" computes parts of the convolution that will contribute to one output pixel, and accumulate the results.
- In each cycle, a weight is broadcasted to all PEs, and the corresponding region of the input feature map is provided to the NFU.

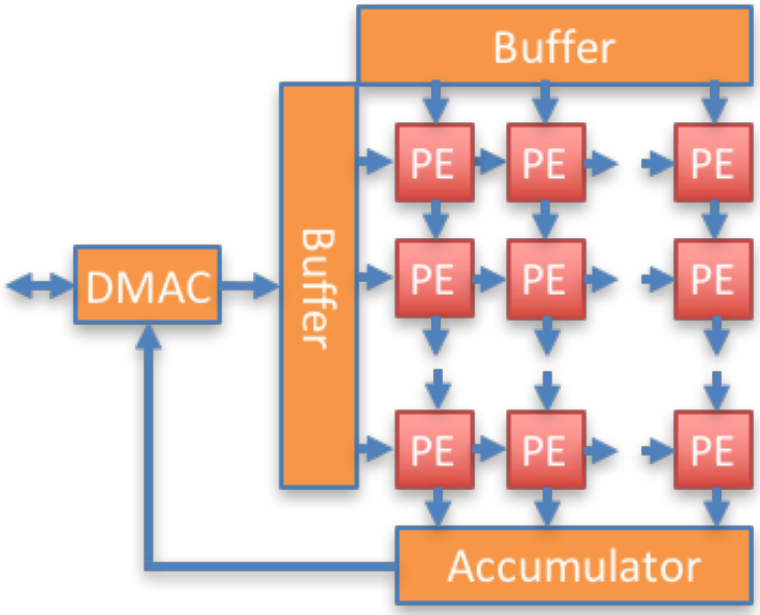
Example of 3x3 convolution on 3x6 input feature map



[1] Z. Du, et. al., "ShiDianNao: Shifting Vision Processing Closer to the Sensor," 2015.

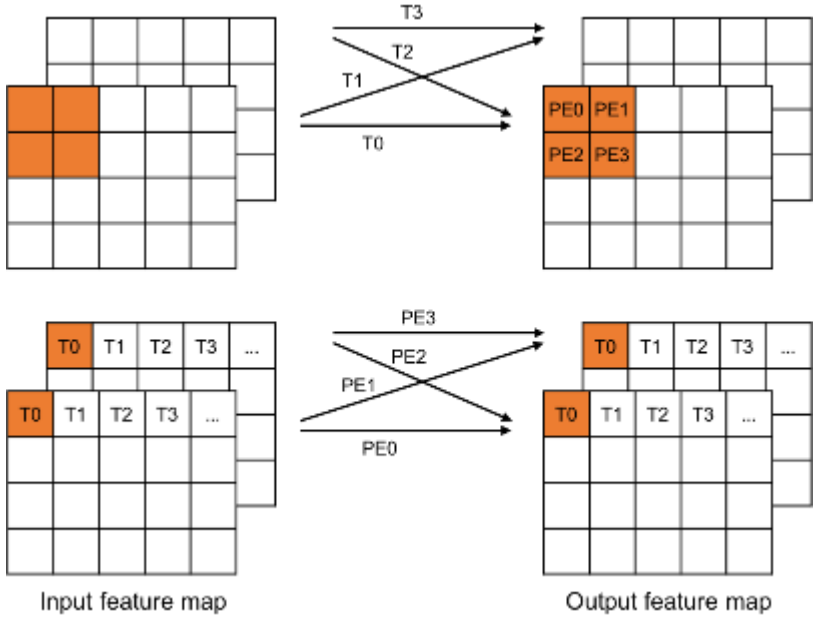
Squeezelerator: Hybrid WS/OS (Berkeley, Samsung)

- Squeezelerator: Supports hybrid WS and OS data flow: up to 6x reduction in energy over OS, WS
  - Determines execution flow statically based on trained weights (one time setup cost per network)



Kwon, Kiseok, Alon Amid, Amir Gholami, Bichen Wu, Krste Asanovic, and Kurt Keutzer. "Co-design of deep neural nets and neural net accelerators for embedded vision applications." In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pp. 1-6. IEEE, 2018.

Output Stationary



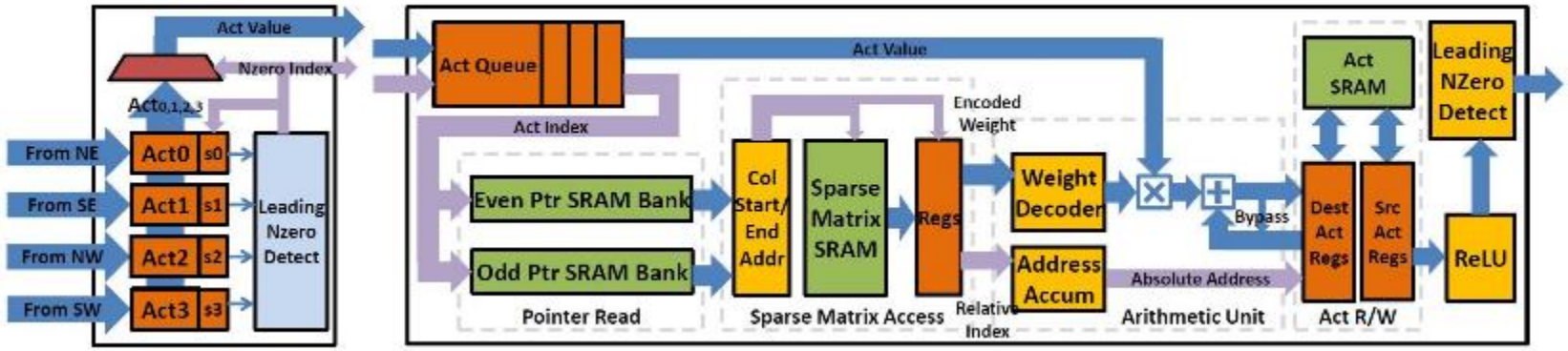
Weight Stationary

Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, datatypes
  - Economical sizes of register files; sufficient memory size and hierarchy
- PE—PE dataflow
  - Hybrid OS/WS dataflow approach
- Other special purpose hardware
  - Support for static compression of weights, sparsity
  - Analog
  - Memory-based computing

HW Support for Compressing Weights

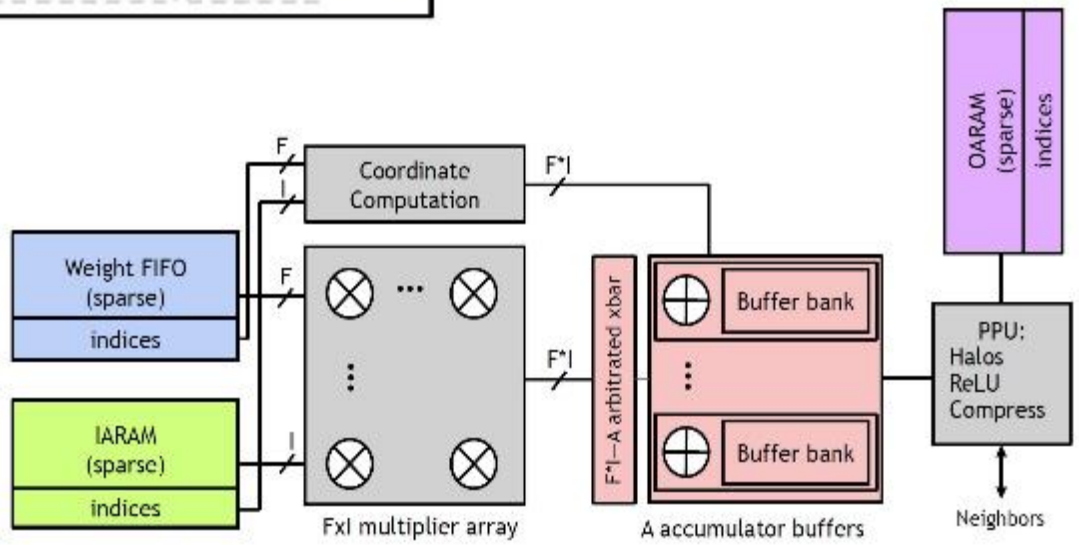
- Typical features include support for weight sparsity, weight quantization, weight sharing, Huffman coding of weights, dynamic activation sparsity, relative indexing, zero detect.



Angshuman Parashar et al., "SCNN: An Accelerator for Compressed-sparse Convolutional Neural Networks". arXiv:1708.04485v1

Song Han et al., "EIE: Efficient Inference Engine on Compressed Deep Neural Network", arXiv:1602.01528v2

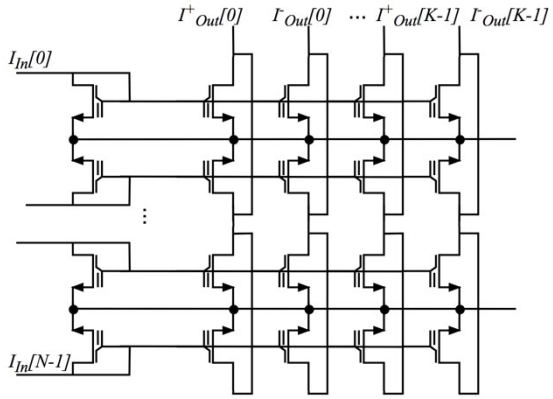
131K weights compressed storage per PE.
Each PE stores part of network in RAM.



Analog Computing: Mythic

- In-memory evaluation of matrix-vector multiply.
- 1-2 orders of magnitude power reduction.
- 40nm embedded Flash process, 8-bit accuracy.
- 5MB on-chip. Claims 0.5pJ/MAC vs. 10-25 pJ/MAC digital.

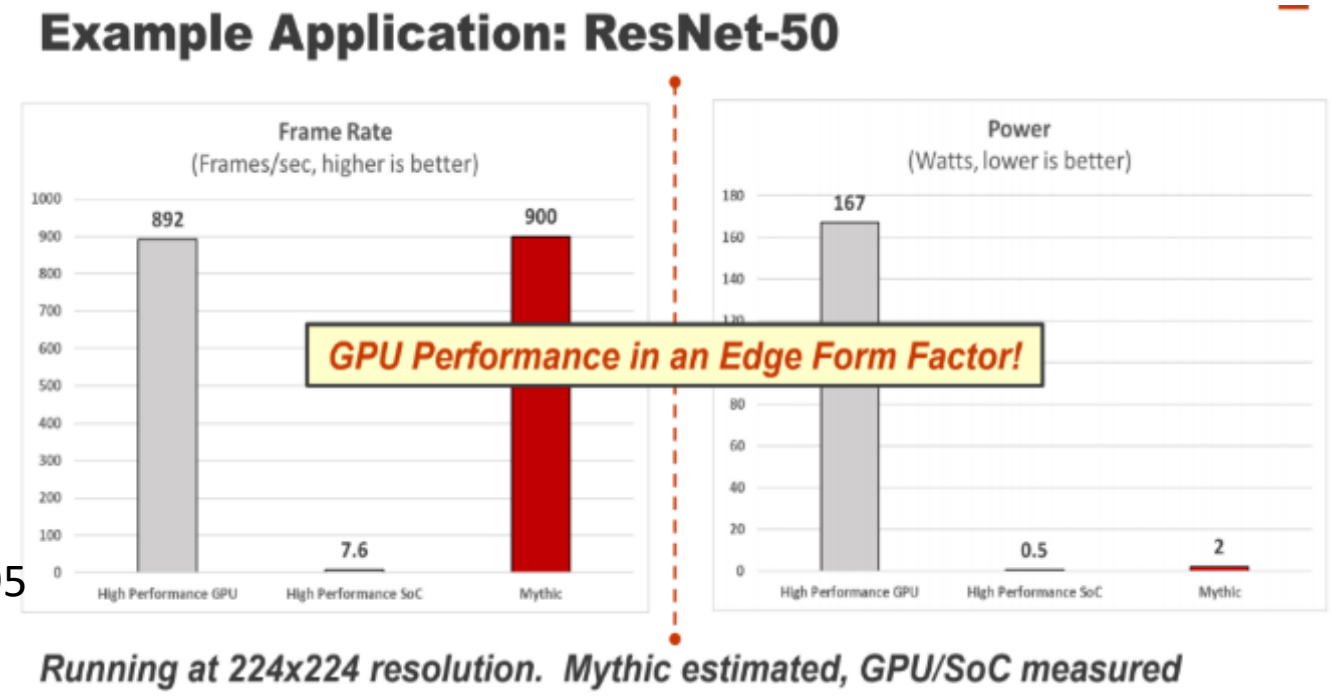
$$I_{Out,j} = \sum_{i=0}^{N} I_{in,i} w_{i,j}$$



Issues:

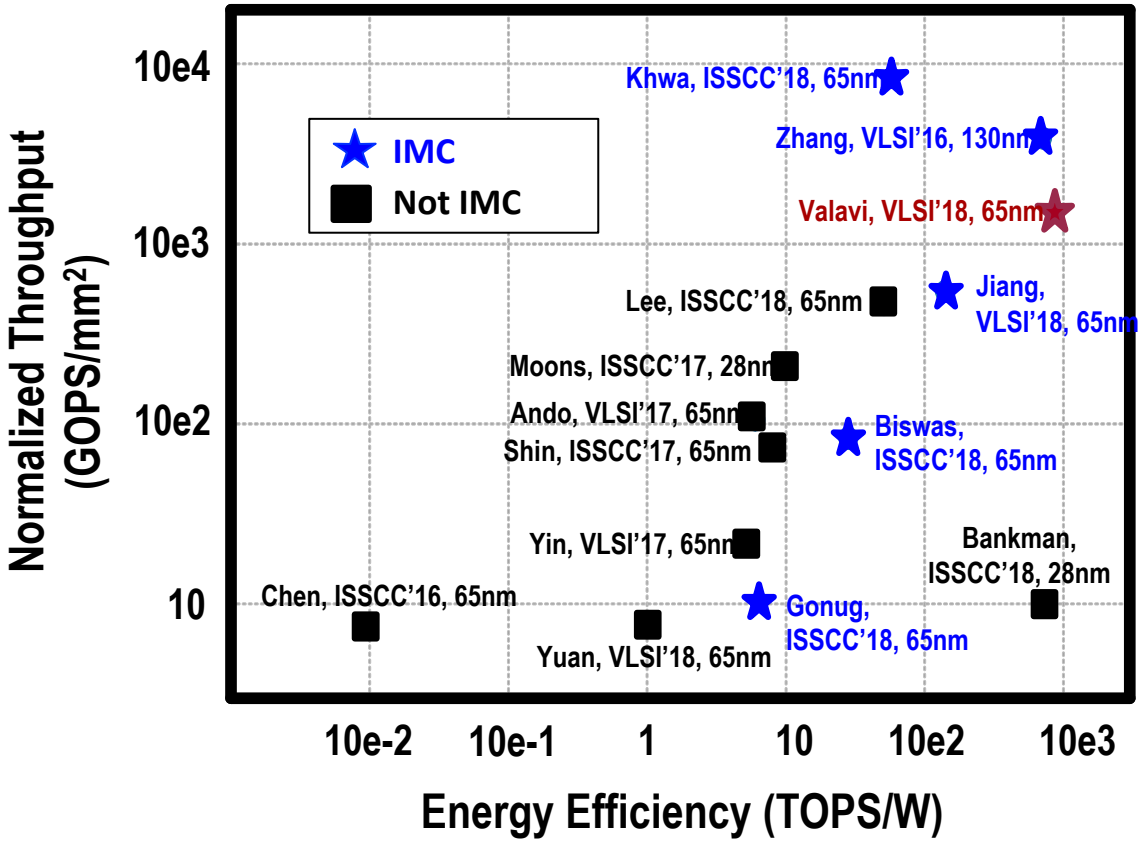
- Sensitivity to process, voltage, temp.
- Embedded Flash process is typically several nodes behind.
- Less benefit from process shrink over time.

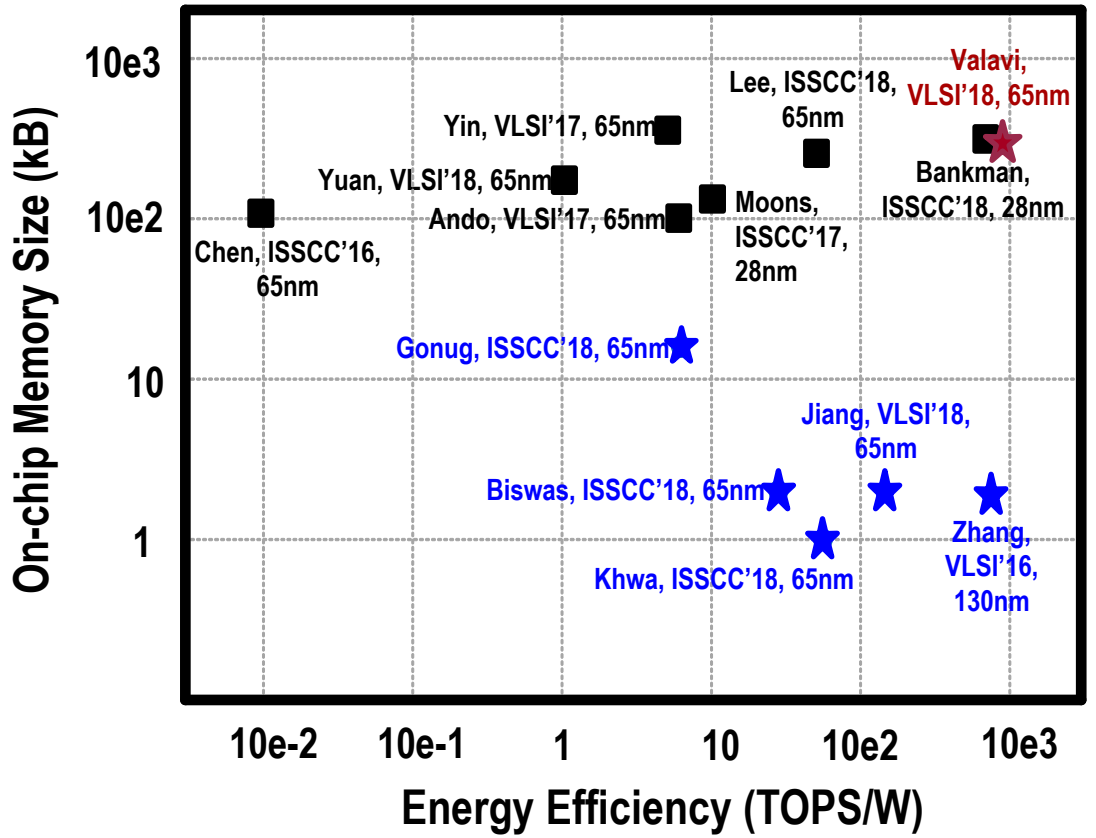
Batch size=1.
Tesla T4 GPU does 4,395 fps @ b=128.



Where does IMC stand today?

- **Potential for 10× higher efficiency & throughput**
- **Limited scale, robustness, configurability**





Naveen Verma, Princeton

"Ready to move from research to R&D."

Key Architectural Decisions

- NN to Accelerator Mapping strategy
  - Layered execution: double buffered
- Number of PE's and their function
  - Minimal PE consisting of MAC unit
- On-chip memory hierarchy, sizes, datatypes
  - Economical sizes of register files; sufficient memory size and hierarchy
- PE—PE dataflow
  - Hybrid OS/WS dataflow approach
- Low hanging fruit is additional memory and arithmetic support for low (1-4 bit) data types

First Summary

- NN accelerators have potential to give 10-100x reduction in latency and energy to Deep Neural Network computations
- But, both DNN design and NN accelerator design are progressing so quickly, the two sub-areas are not keeping up with each other
- Application constraints for NN accelerators are quite different, so need to focus:
  - Most important application constraint, accuracy, is often implicit or misunderstood
  - Cloud or client?
  - AI sub-area : Computer vision, speech, natural language processing? Talk was all CV
- For accelerators for inference in mobile/at-the-edge NN, there are many familiar architectural choices to be made, and given up-to-date DNN models traditional data-driven architectural analysis can drive them
  - Factors of 10x hinge on making these correct choices in the light of application constraints and DNN characteristics: PE, memory hierarchy uppermost, dataflow
  - Anything new?: support for low-bit precisions (1-4 bit) datatypes is low hanging fruit
- Given a good NN accelerator architecture that is tuned to support a family of nets, presuming quantization, further DNN optimization currently might net 2X
- Much bigger gains (10x) if we can tune the NN accelerator to a *particular* application and co-design the DNN and NN accelerator for the application (e.g. eliminating 3x3 convolutions altogether)

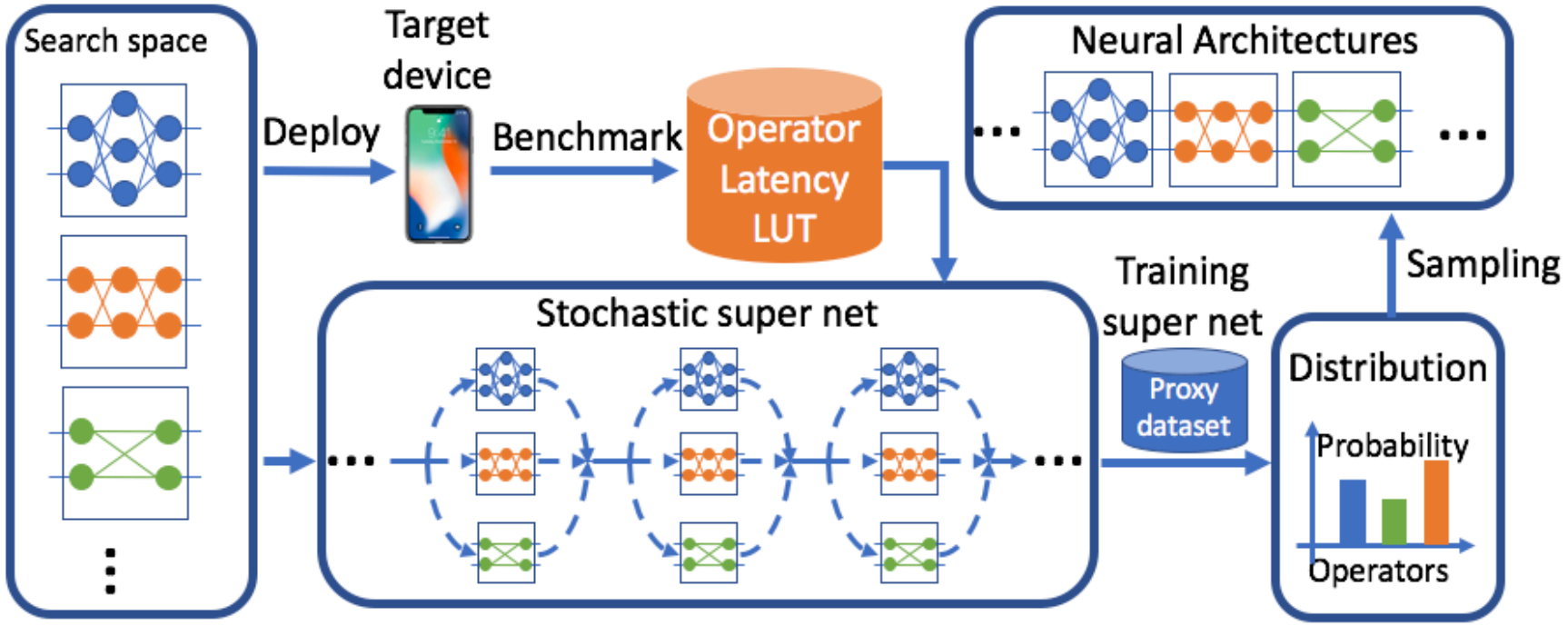
Outline

- Applications and their characteristics
- Determining key NN Accelerator architectural elements
- How much further can we improve NN accelerators when we co-design them with DNNs?

Finding the right DNN model just got a lot easier!!
DNAS: Differentiable Neural Architecture Search

Differentiable Neural Architecture Search:

- Extremely fast: 8 GPUs, 24 hours
- Optimize for actual latency

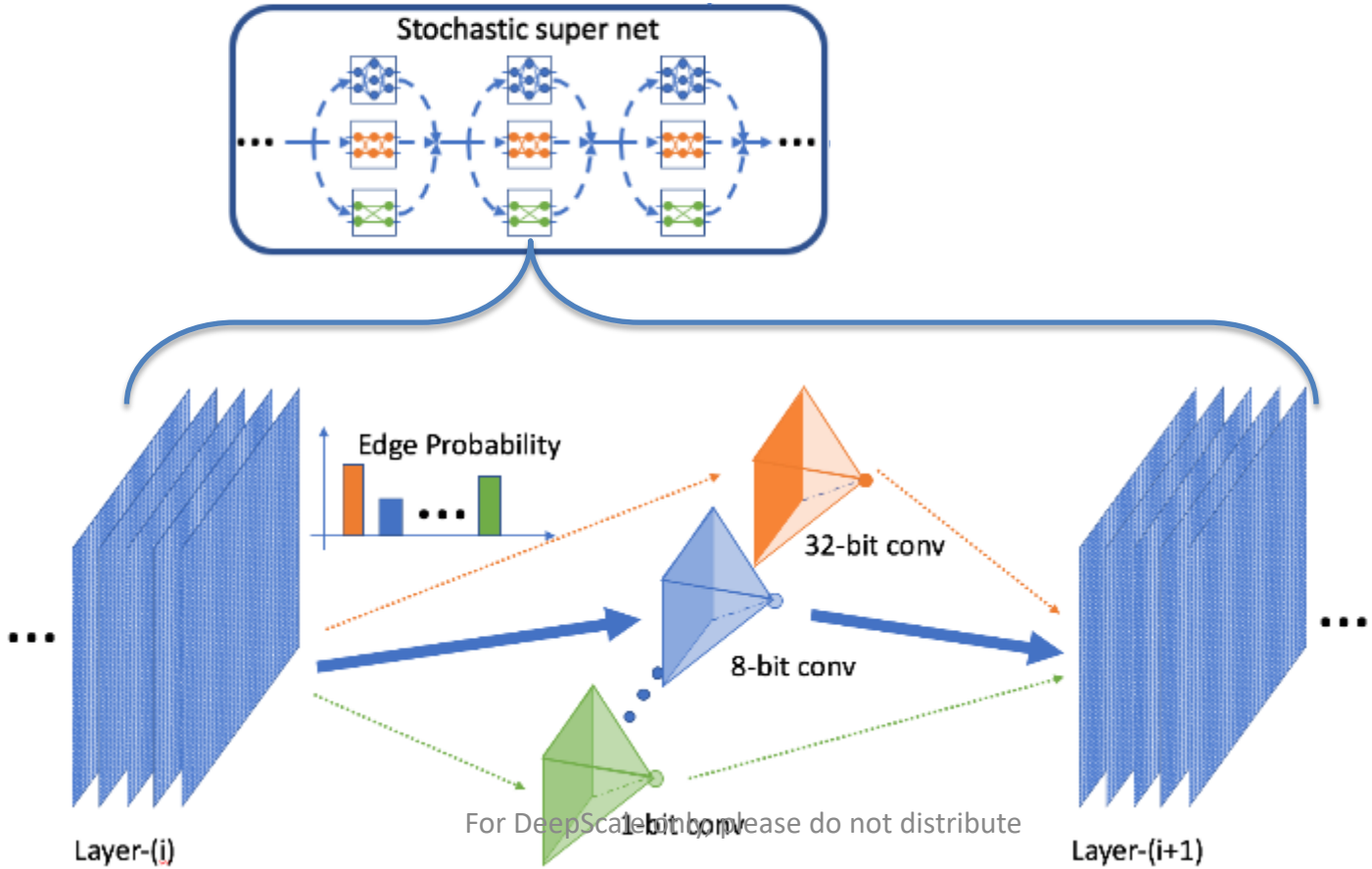


CVPR Oral 2019

Wu, Bichen, Xiaoliang Dai, Peizhao Zhang, Yanghan Wang, Fei Sun, Yiming Wu, Yuandong Tian, Peter Vajda, Yangqing Jia, and Kurt Keutzer. "FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search." *arXiv preprint arXiv:1812.03443* (2018).

DNAS for mixed precision quantization search

- Quantizing different layers of a ConvNet to different precisions
- Candidate operators are quantized convolutions

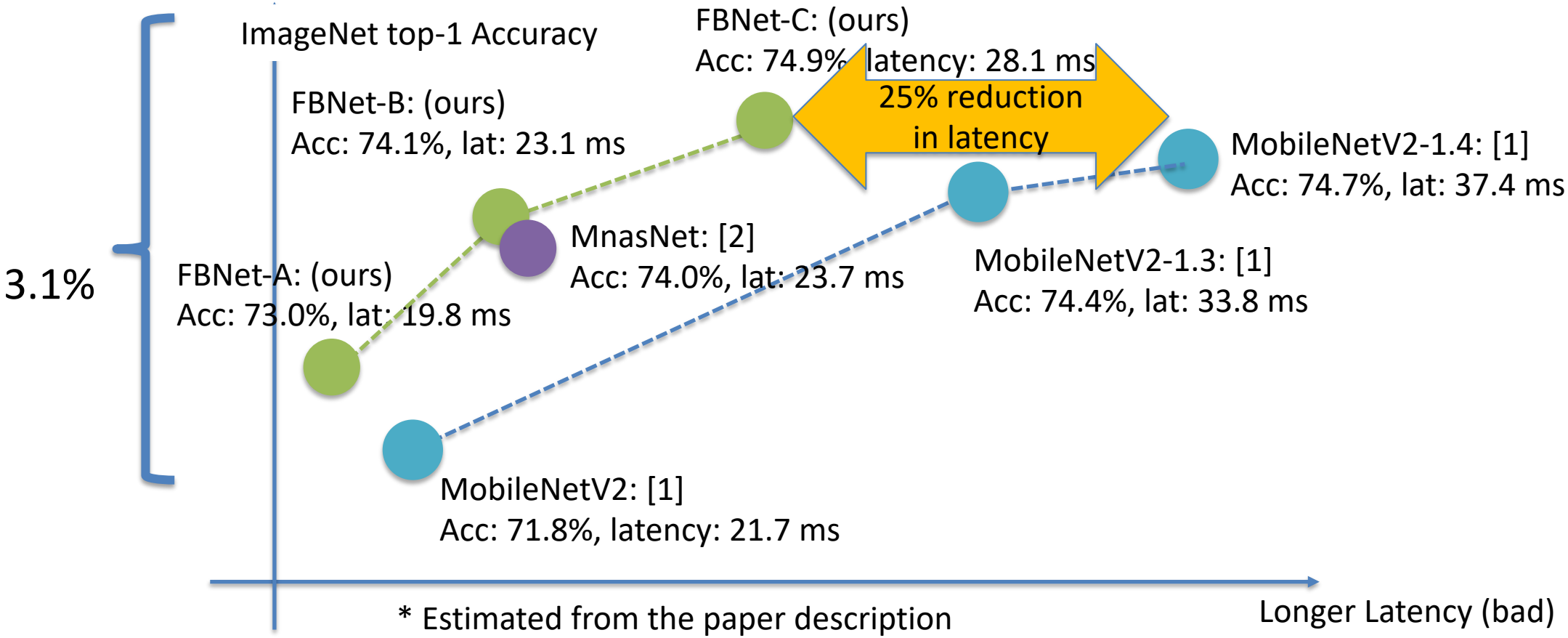


Quantization Can Save 12X: Inception-V3 on ImageNet

Method	w-bit	a-bit	Top1	W-Comp	Size(MB)
Baseline	32	32	77.45	1.00×	91.0
Integer-Only [6]	8	8	75.40	4.00×	22.8
Integer-Only [6]	7	7	75.00	4.57×	19.9
Deep Comp. [7]	3	–	75.10	10.41×	9.36
HAQ [8]	–	–	**75.30**	10.57×	9.22
Direct	2	4	69.76	15.88×	5.73
HGQ [1]	2	4	**75.45**	12.04×	**7.56**

Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer,
HGQ: Hessian Guided Quantization of Neural Networks with Mixed-Precision

FBNet Family in context (Latency)
Tools Beating the Best Human DNN Designers



[1] Sandler, Mark, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks." CVPR18

[2] Tan, Mingxing, et al. "Mnasnet: Platform-aware neural architecture search for mobile." *arXiv preprint arXiv:1807.11626* (2018).

Design Study 1:
A11 vs Snapdragon 835

Apple A11

- Big: 2 ARMv8 @ 2.5 GHz
- Little: 4 ARMv8 @ 1.4 GHz
- Vectorization: 4-wide 32-bit MAC
- LPDDR4x memory (30 GB/s)
- GPU + Neural Processing Engine

Snapdragon 835

- Big: 4 ARMv8 @ 2.4 GHz
- Little: 4 ARMv8 @ 1.9 GHz
- Vectorization: 4-wide 32-bit MAC
- LPDDR4x memory (30 GB/s)
- Adreno 540 GPU

DNAS for device-aware search

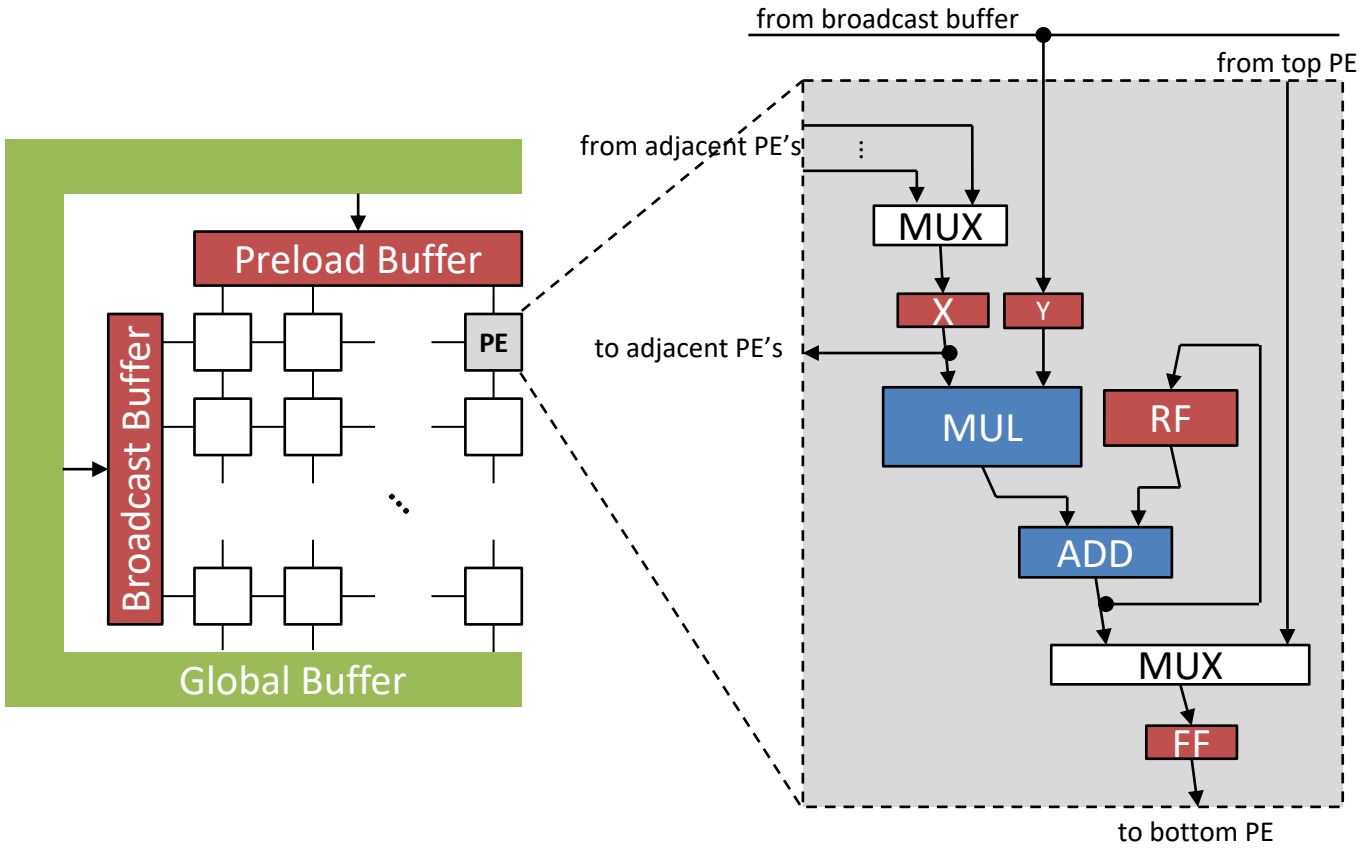
- For different targeted devices, both DNASNets achieve similar accuracy.
- However, per target DNN optimization yield 20-25% reduction in latency

NET	Latency on iPhoneX	Latency on Samsung S8	Top-1 acc
DNAS-iPhoneX	19.84 ms	23.33 ms (20% slower)	73.20%
DNAS-S8	27.53 ms (25% slower)	22.12 ms	73.27%

Design Study 2: Squeezelerator Collaboration with Samsung (Kwon)

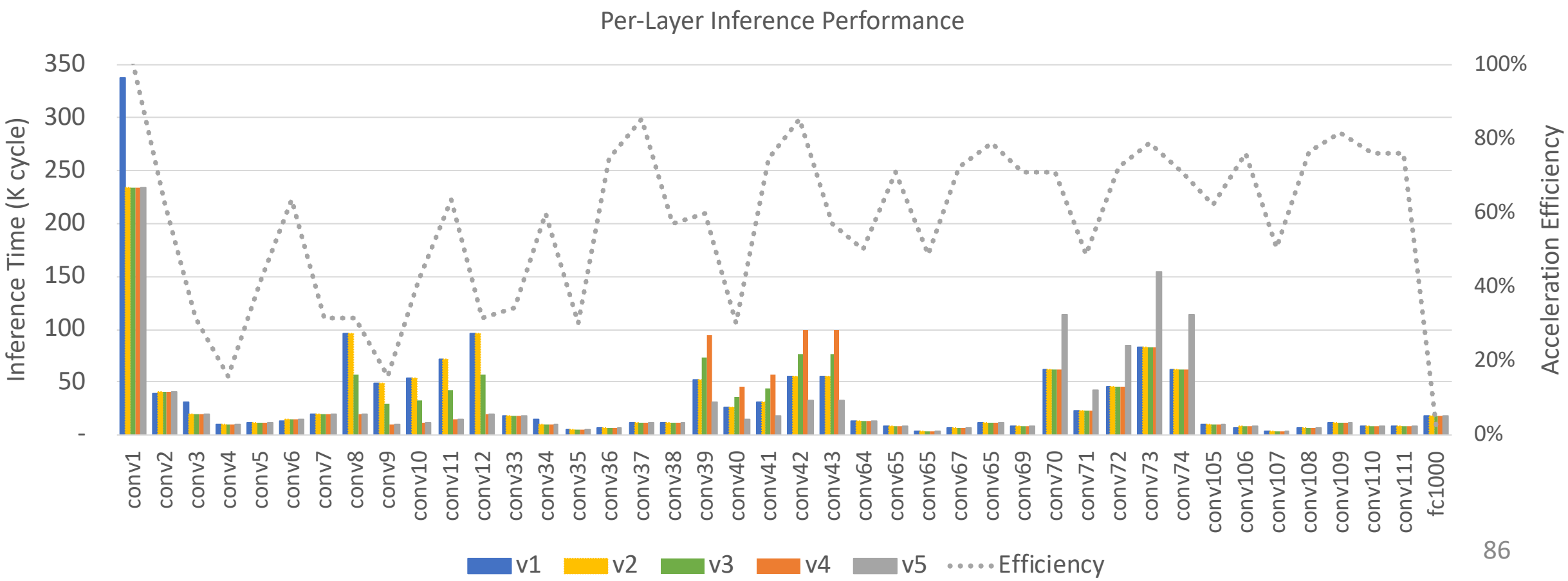
- Squeezelerator evolved to support both OS and WS dataflow modes.
- Initial results improved accelerator performance on MobileNets v1 by 6x

Kwon, Kiseok, Alon Amid, Amir Gholami, Bichen Wu, Krste Asanovic, and Kurt Keutzer. "Co-Design of Deep Neural Nets and Neural Net Accelerators for Embedded Vision Applications." DAC 2018. To appear IBM Journal of Research and Development.
Also, arXiv:1804.10642 (2018).



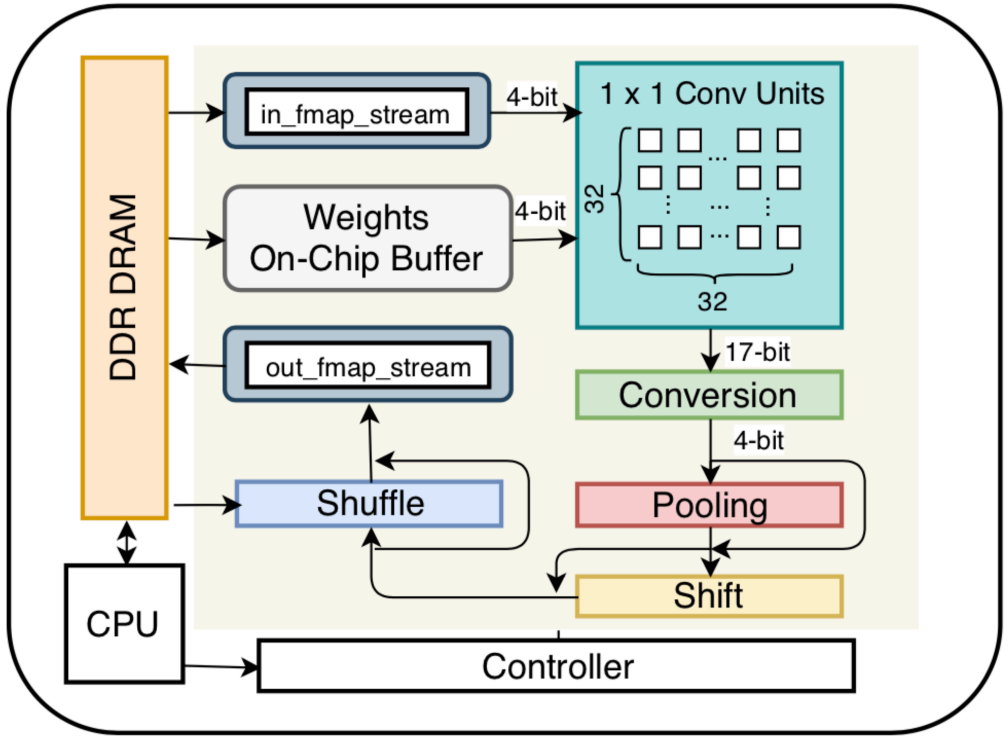
Subsequent DNN Optimization

- After Squeezelerator was designed to optimize SqueezeNet, MobileNet, further optimizing SqueezeNext to Squeezelerator improved energy efficiency/latency by (only) 30%



Design Study 3: DiracDeltaNet & Synetgy Accelerator [1]

- Collaboration with Xilinx
- Xilinx Zynq ZU3EG
  - Relatively weaker (than GPU) support for linear algebra
  - Very flexible
  - Excellent support for bit-level operations
  - Excellent support for fixed-point quantization
- Developed Synetgy NN Accelerator



Synetgy accelerator architecture

Yang, Yifan, Qijing Huang, Bichen Wu, Tianjun Zhang, Liang Ma, Giulio Gambardella, Michaela Blott et al. "Synetgy: Algorithm-hardware co-design for convnet accelerators on embedded fpgas." In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 23-32. ACM, 2019.

Co-Design DNN: DiracDeltaNet

- Dramatically simplified operation set to better match FPGA

- ShuffleNetV2 [1]
  - 1x1 conv
  - 3x3 conv stride=2
  - 3x3 depth-wise conv stride=1
  - 3x3 depth-wise conv stride=2
  - 3x3 max-pooling
  - Shuffle and concatenation

- DiracDeltaNet
  - Eliminated spatial convolutions altogether
  - 1x1 conv
  - 2x2 max-pooling
  - Shift [2]
  - Shuffle and concatenation

[1] Ma, Ningning, et al. "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design." *arXiv preprint arXiv:1807.11164* (2018).
[2] Wu, Bichen, Alvin Wan, Xiangyu Yue, Peter Jin, Sicheng Zhao, Noah Golmant, Amir Gholaminejad, Joseph Gonzalez, and Kurt Keutzer. "Shift: A Zero FLOP, Zero Parameter Alternative to Spatial Convolutions

Comparison with Previous Works

	Platform	Framerate	Top-1 Acc	Precision	Energy/ Frame (J)
VGG16 [1]	Zynq 7Z020	5.7	67.72%	8-8b	0.526
VGG-SVD [2]	Zynq 7Z045	4.5	64.64%	16-16b	0.666
VGG16 [3]	Stratix-V	3.8	66.58%	8-16b	5.026
Ours	Zynq ZU3EG	**66.3**	**68.30%**	**4-4b**	**0.083**

- 11.6x faster (among FPGA accelerators whose top-1 accuracy is higher than 60%)
- 6.3x more power efficient

[1] Guo, K., Han, S., Yao, S., Wang, Y., Xie, Y. and Yang, H. Software-Hardware Codesign for Efficient Neural Network Acceleration. IEEE Micro, 37 (2). 18-25.
[2] Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., Yu, J., Tang, T., Xu, N., Song, S., Wang, Y. and Yang, H. Going Deeper with Embedded {FPGA} Platform for Convolutional Neural Network, 2016, 26-35.
[3] Suda, N., Chandra, V., Dasika, G., Mohanty, A., Ma, Y., Vrudhula, S.B.K., Seo, J.S. and Cao, Y. Throughput-Optimized OpenCL-based {FPGA} Accelerator for Large-Scale Convolutional Neural Networks, 2016, 16-25.

Summary

- NN accelerators have potential to give 10-100x reduction in latency and energy to Deep Neural Network computations
- But, both DNN design and NN accelerator design are progressing so quickly, the two sub-areas are not keeping up with each other
- Application constraints for NN accelerators are quite different, so need to focus:
  - Most important application constraint, accuracy, is often implicit or misunderstood
  - Cloud or client?
  - AI sub-area : Computer vision, speech, natural language processing? Talk was all CV
- For accelerators for inference in mobile/at-the-edge NN, there are many familiar architectural choices to be made, and given up-to-date DNN models traditional data-driven architectural analysis can drive them
  - Factors of 10x hinge on making these correct choices in the light of application constraints and DNN characteristics: PE, memory hierarchy uppermost, dataflow
  - Anything new?: support for low-bit precisions (1-4 bit) datatypes is low hanging fruit
- Given a good NN accelerator architecture that is tuned to support a family of nets, presuming quantization, further DNN optimization currently might net 2X
- Much bigger gains (10x) if we can tune the NN accelerator to a *particular* application and co-design the DNN and NN accelerator for the application (e.g. eliminating 3x3 convolutions altogether)

Extras

The way it was: Dennard Scaling

Practical Strategies for

Power-Efficient Computing Technologies

By Leland Chang, Dennard et al., Proceedings of the IEEE | Vol. 98, No. 2, February 2010

Table 1 Scaling Theory to Maintain Constant Electric Fields in a MOSFET Device. κ is a Dimensionless Scale Factor. From Dennard *et al.* [2]

Device or Circuit Parameter	Scaling Factor
Device dimension t_{ox}, L, W	$1/\kappa$
Doping concentration N_a	κ
Voltage V	$1/\kappa$
Current I	$1/\kappa$
Capacitance $\varepsilon A/t$	$1/\kappa$
Delay time/circuit VC/I	$1/\kappa$
Power dissipation/circuit VI	$1/\kappa^2$
Power density VI/A	1

κ is the dimensional scale factor

The way it is: End of power scaling

Practical Strategies for

Power-Efficient Computing Technologies

By Leland Chang, Dennard et al., Proceedings of the IEEE | Vol. 98, No. 2, February 2010

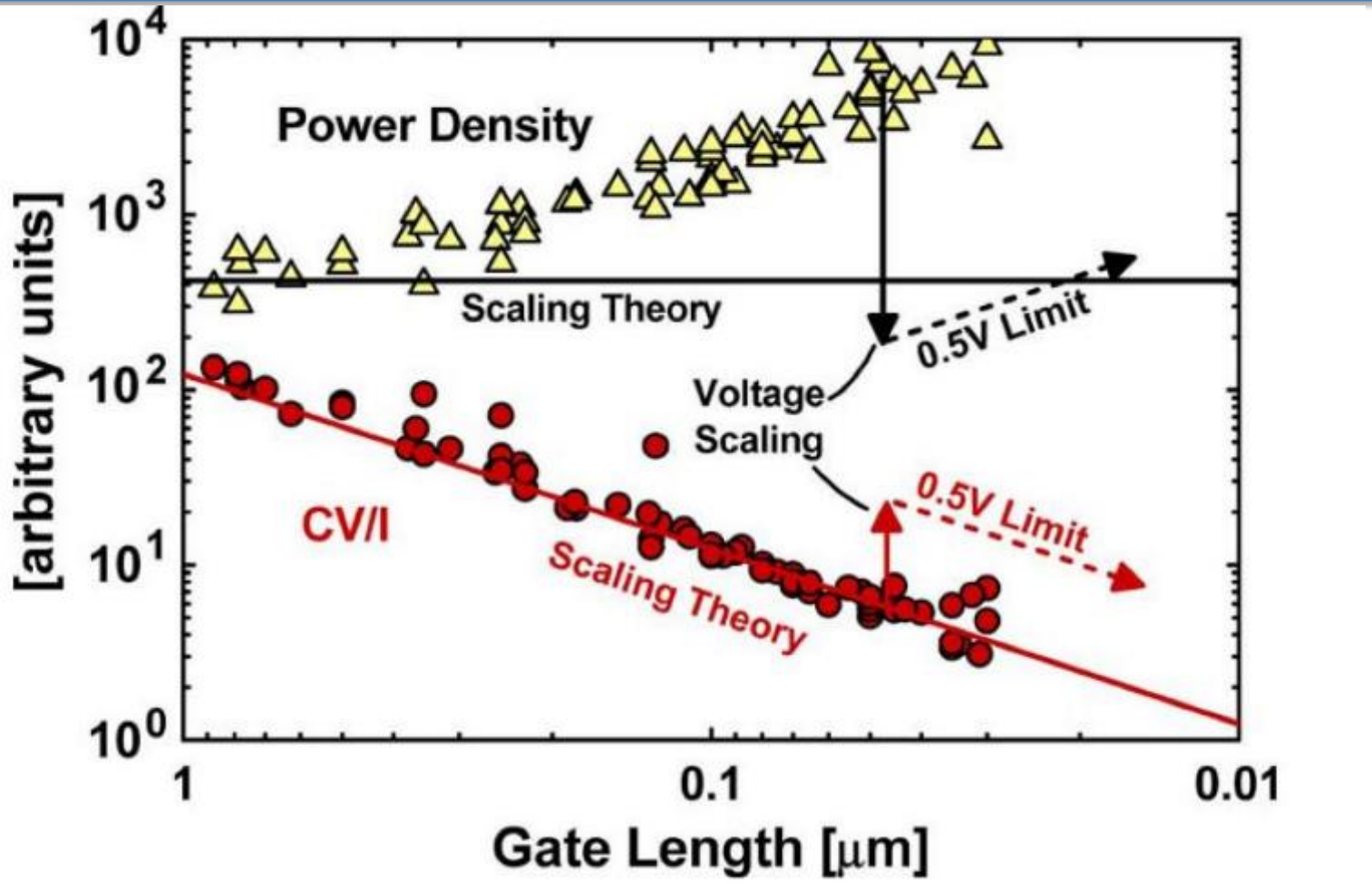
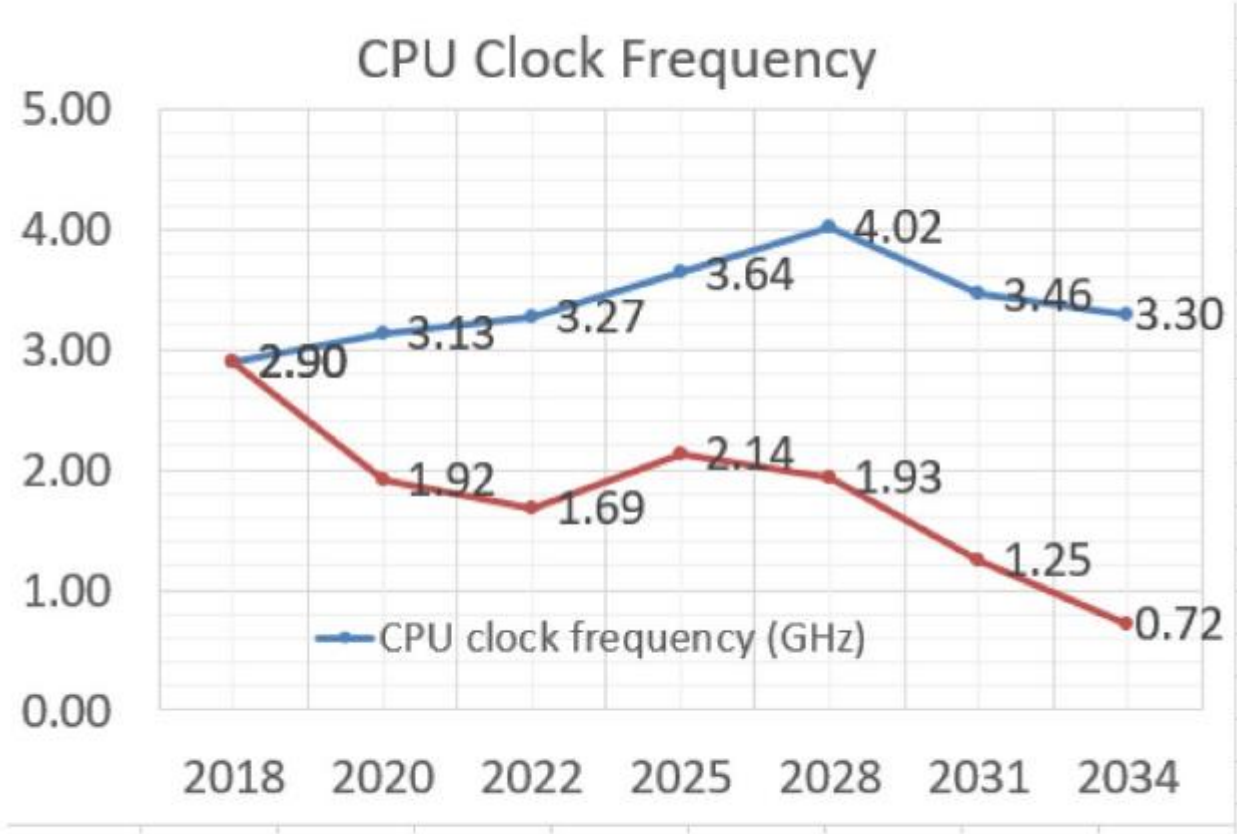


Fig. 4. ***To maintain performance (CV/I) trends, voltage scaling has slowed, which results in dramatic increases in power density.***

The way it is: Limited clock rates



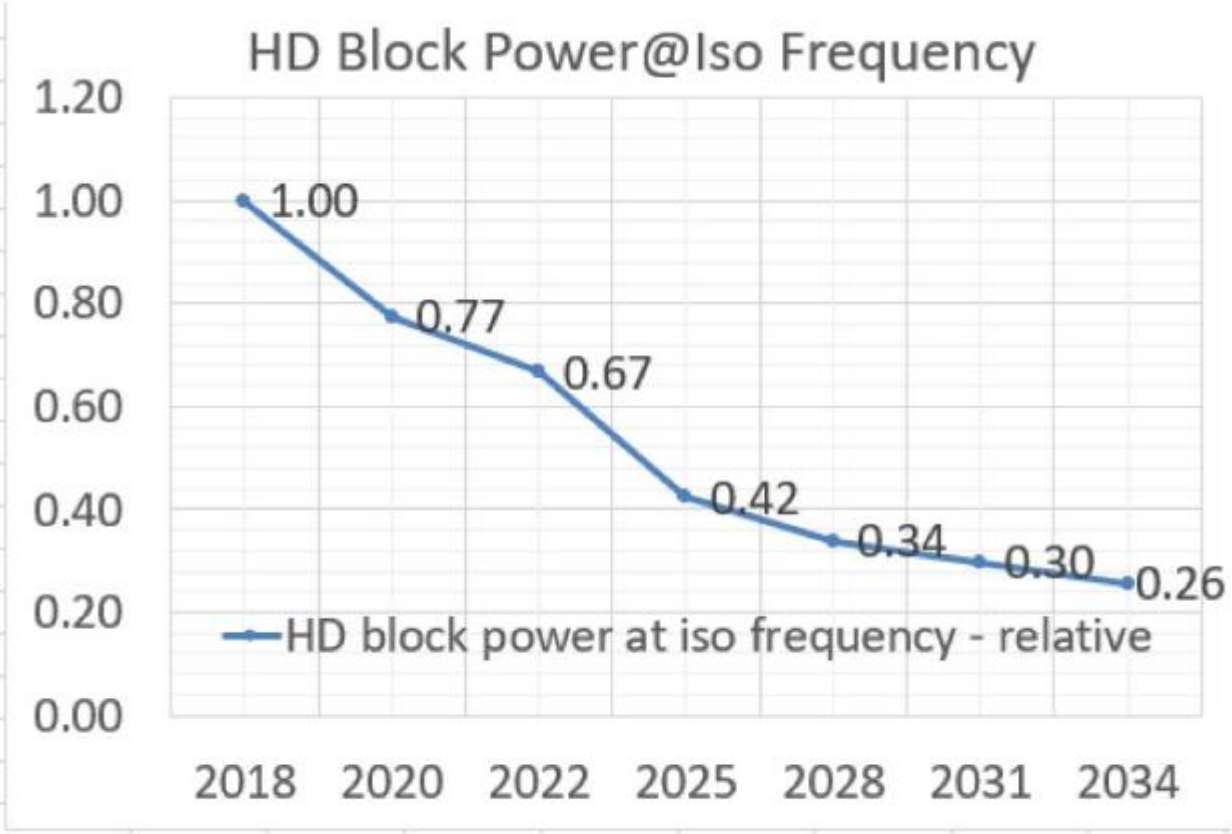


Figure MM-9 CPU clock frequency and power@iso- frequency (ref: 2018) scaling

IEEE INTERNATIONAL ROADMAP FOR DEVICES AND SYSTEMS, 2018 EDITION . Red curve – CPU freq at constant power density.

Consider I/O Bandwidth and Latency as well…

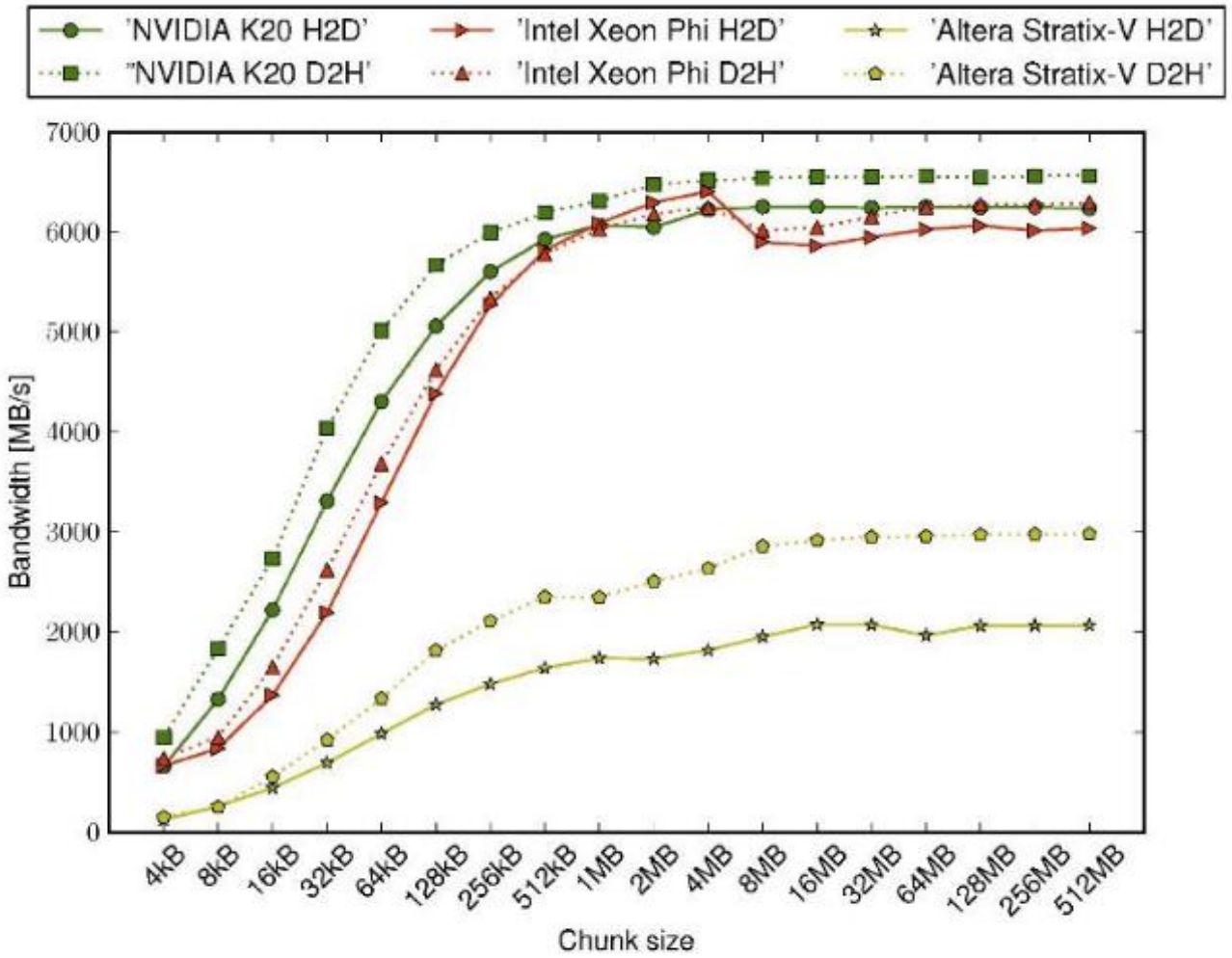


Fig. 1: Measured PCIe host-to-device (H2D) and device-to-host (D2H) bandwidth for the applied co-processor cards.

Analyzing the Energy-Efficiency of Sparse Matrix Multiplication on Heterogeneous Systems: A Comparative Study of GPU, Xe on Phi and FPGA
Heiner Giefers et al., 2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)

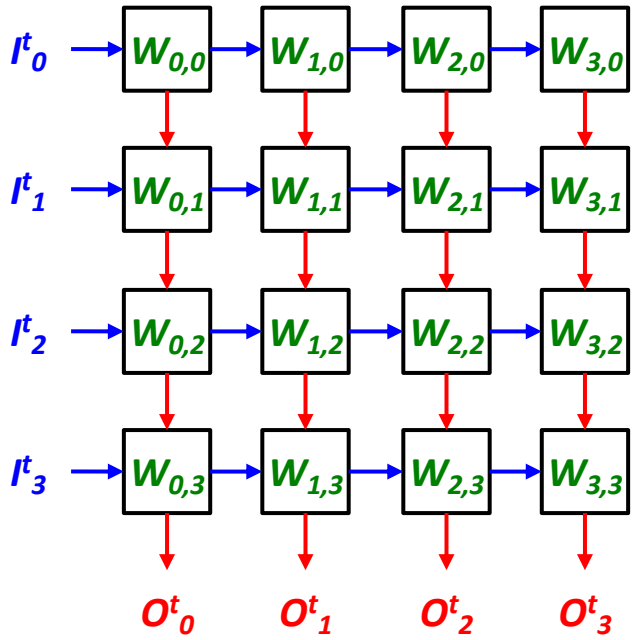
Animated Versions

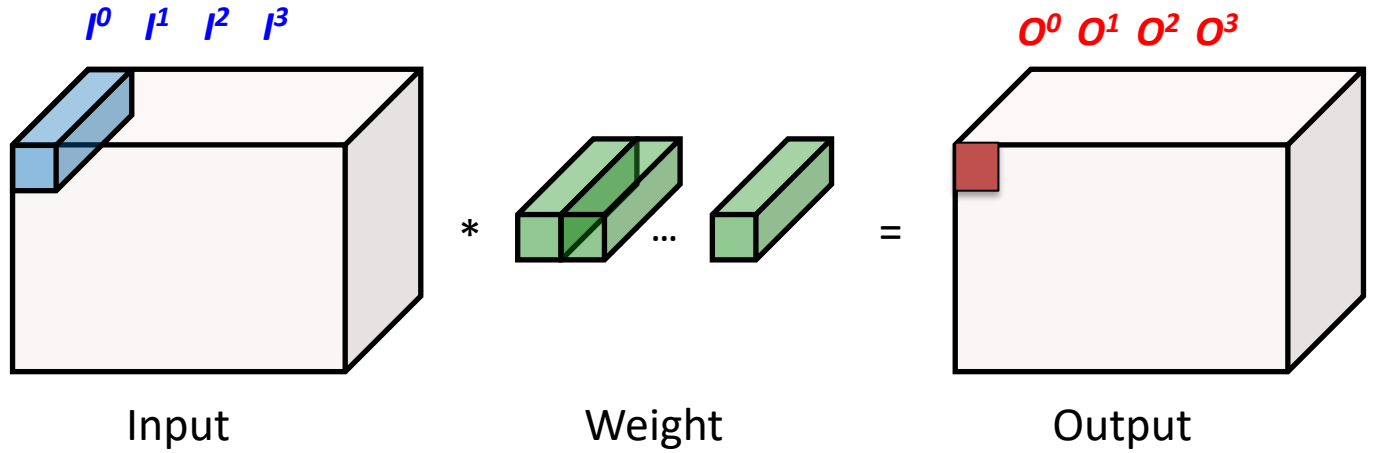
Systolic Array Weight Stationary Example: TPU

- "Matrix Multiply Unit" performs general matrix-vector multiplications.
- The weight matrix is preloaded in the PE Array.
- A stream of input activation vectors is passed to each column of the array.
- Partial sums of PEs are vertically propagated

Output = Weight-Matrix x Inputs

$$\begin{bmatrix} O_0^t \\ \vdots \\ O_{k-1}^t \end{bmatrix} = \begin{bmatrix} W_{0,0} & \cdots & W_{0,c-1} \\ \vdots & \ddots & \vdots \\ W_{k-1,0} & \cdots & W_{k-1,c-1} \end{bmatrix} \begin{bmatrix} I_0^t \\ \vdots \\ I_{c-1}^t \end{bmatrix}$$



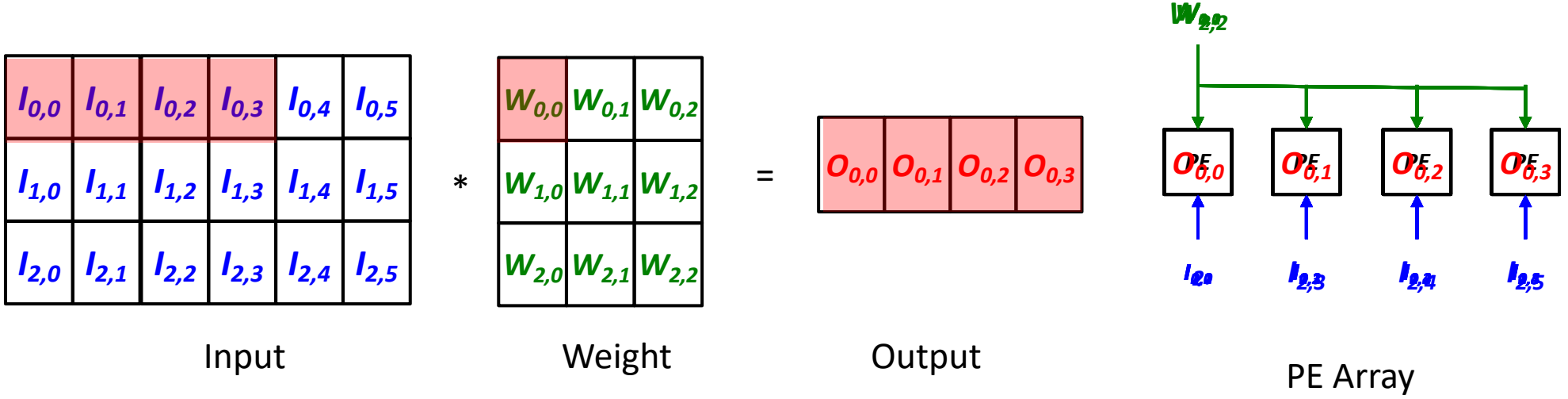


[1] N. Jouppi, et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit," 2017.

Systolic Array Output Stationary Example: ShiDianNao

- Each PE in "Neural Functional Unit" computes parts of the convolution that will contribute to one output pixel, and accumulate the results.
- In each cycle, a weight is broadcasted to all PEs, and the corresponding region of the input feature map is provided to the NFU.

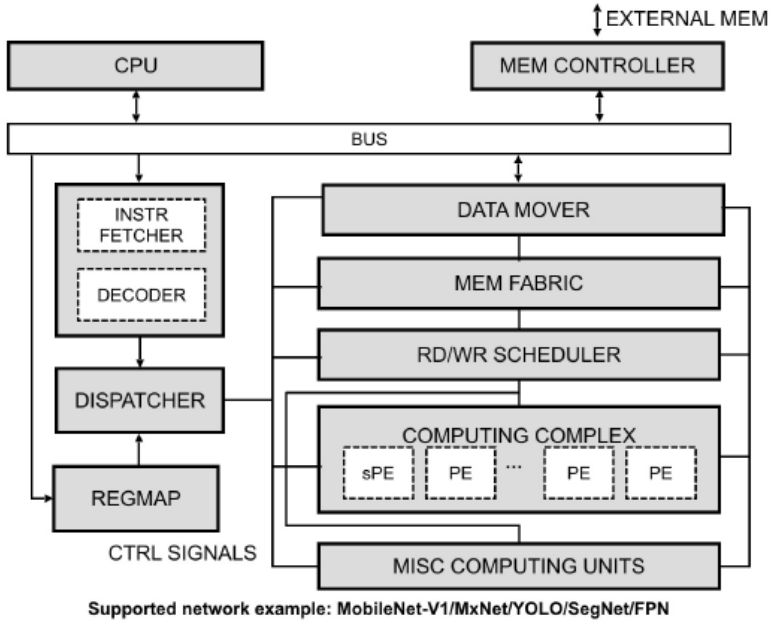
Example of 3x3 convolution on 3x6 input feature map



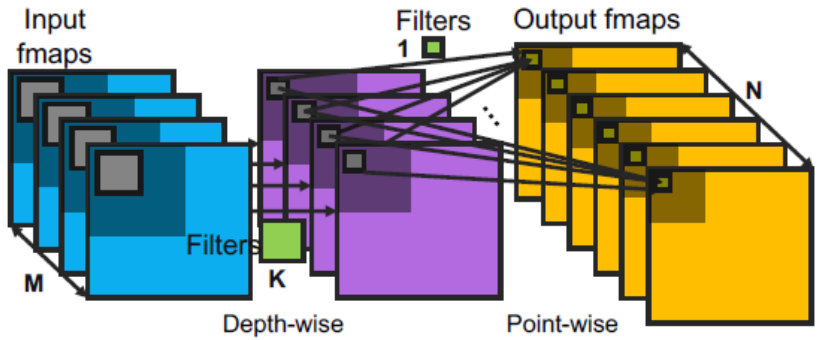
[1] Z. Du, et. al., "ShiDianNao: Shifting Vision Processing Closer to the Sensor," 2015.

Co-Design with DeepPhi

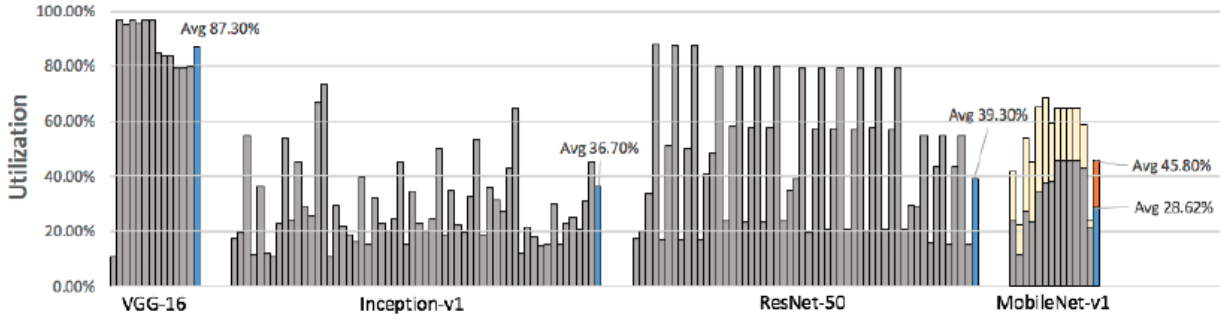
DPU [DeePhi, 2018]



(a) DeePhi's DPU Architecture for MobileNet

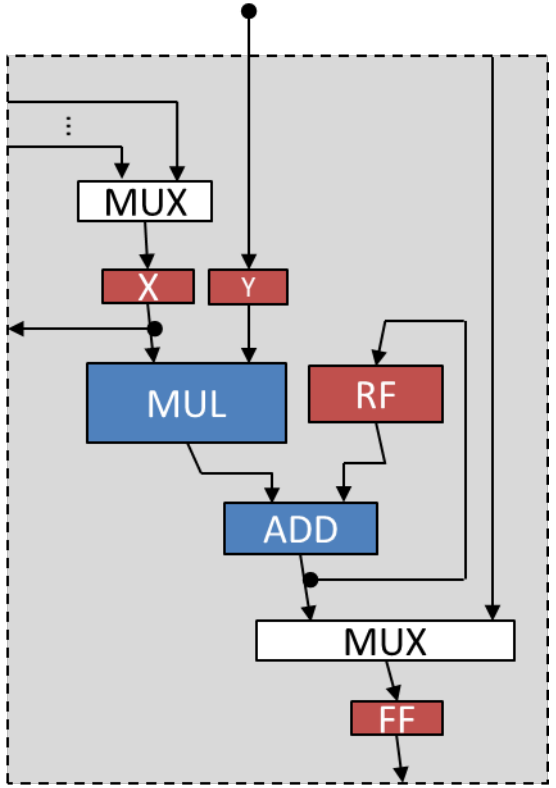


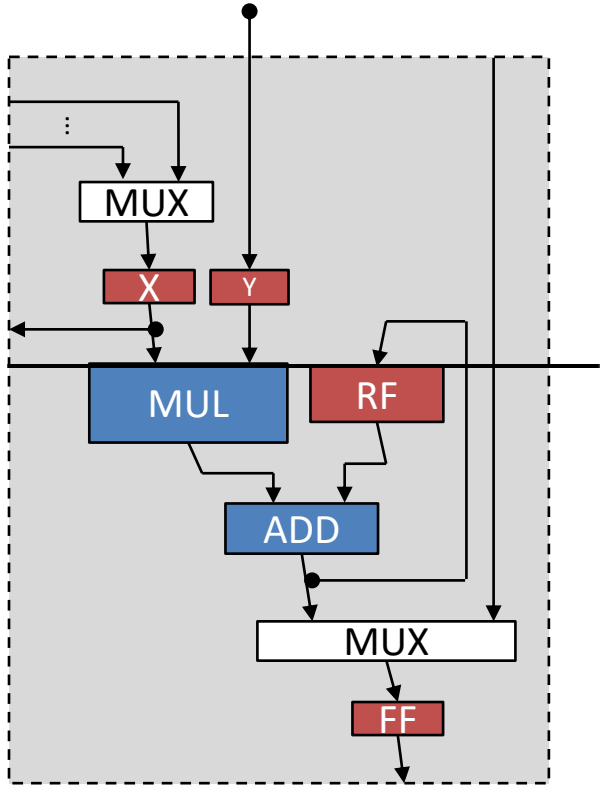
(b) Fused DP-convolution and PW-convolution

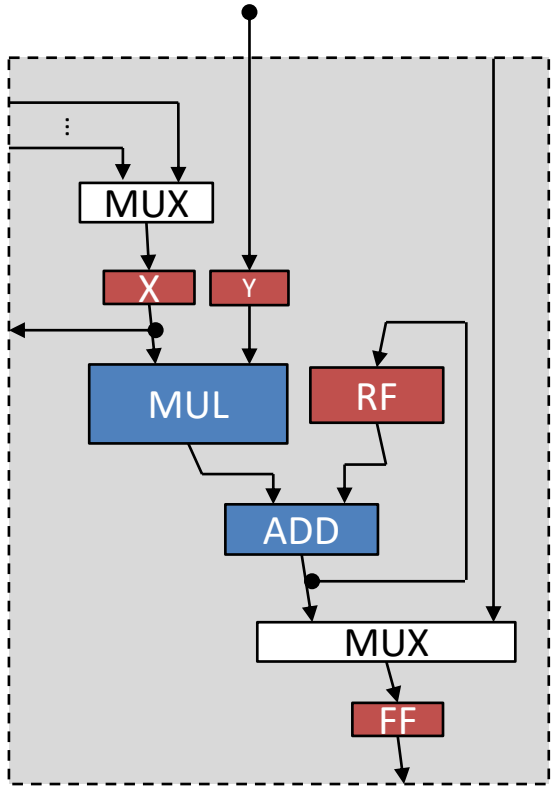


(c) Result on Xilinx ZU9 FPGA: increased the utilization from 28.6% to 45.8%

MAC Unit


MUX
X
Y
MUL
RF
ADD
MUX
FF


MUX
X
Y
MUL
RF
ADD
MUX
FF


MUX
X
Y
MUL
RF
ADD
MUX
FF

Added Material

EyeRiss V2: Flexible NoC, multiple forms of data re-use

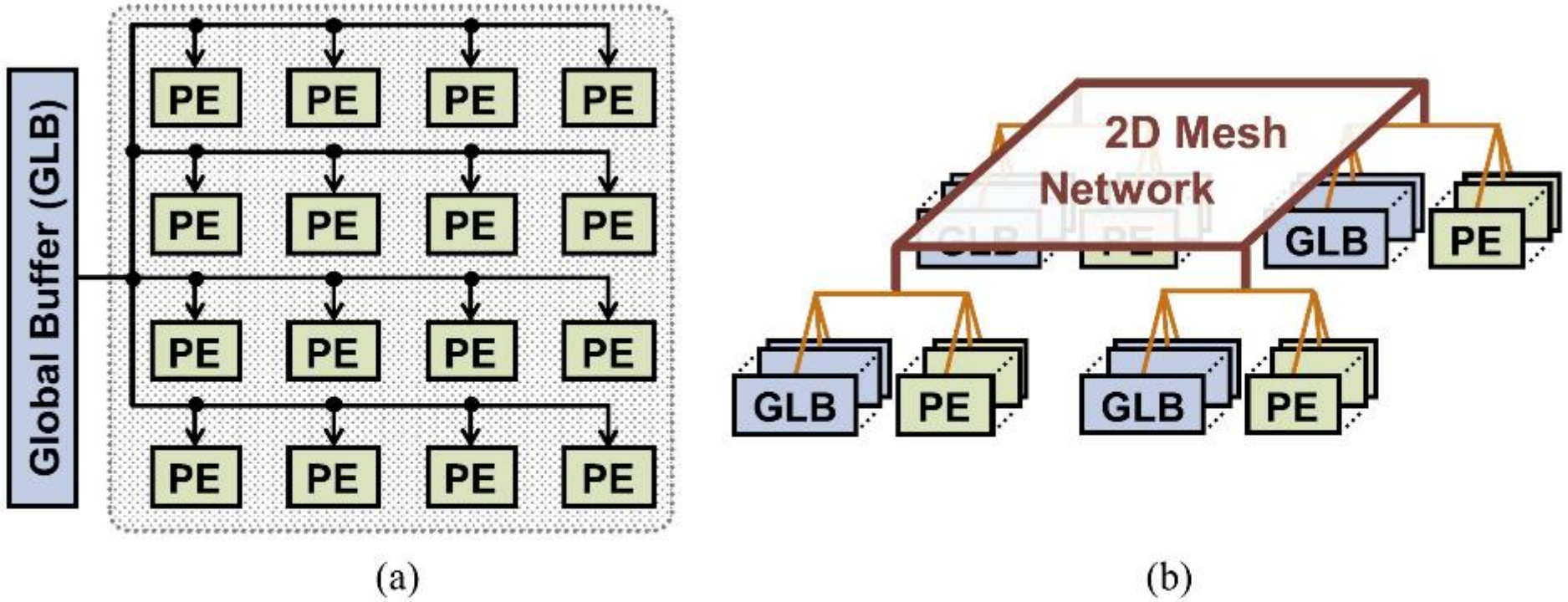
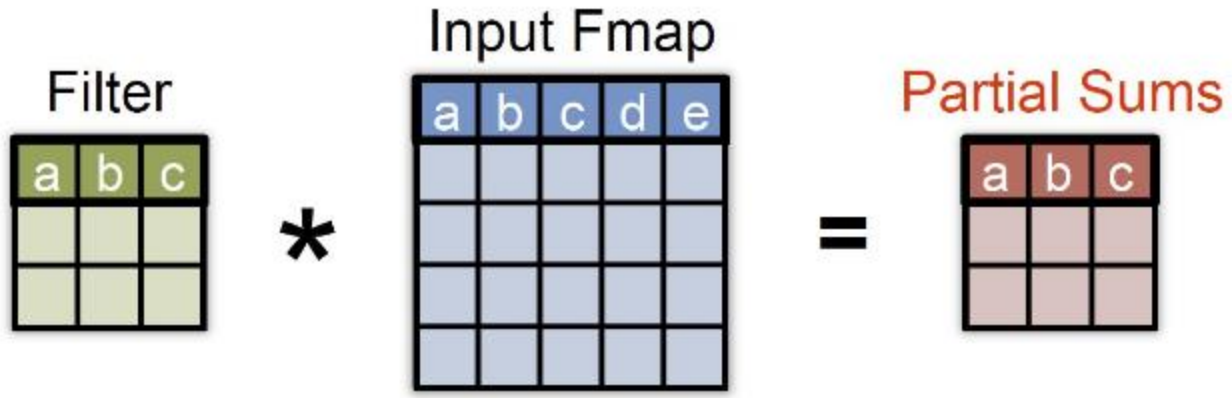


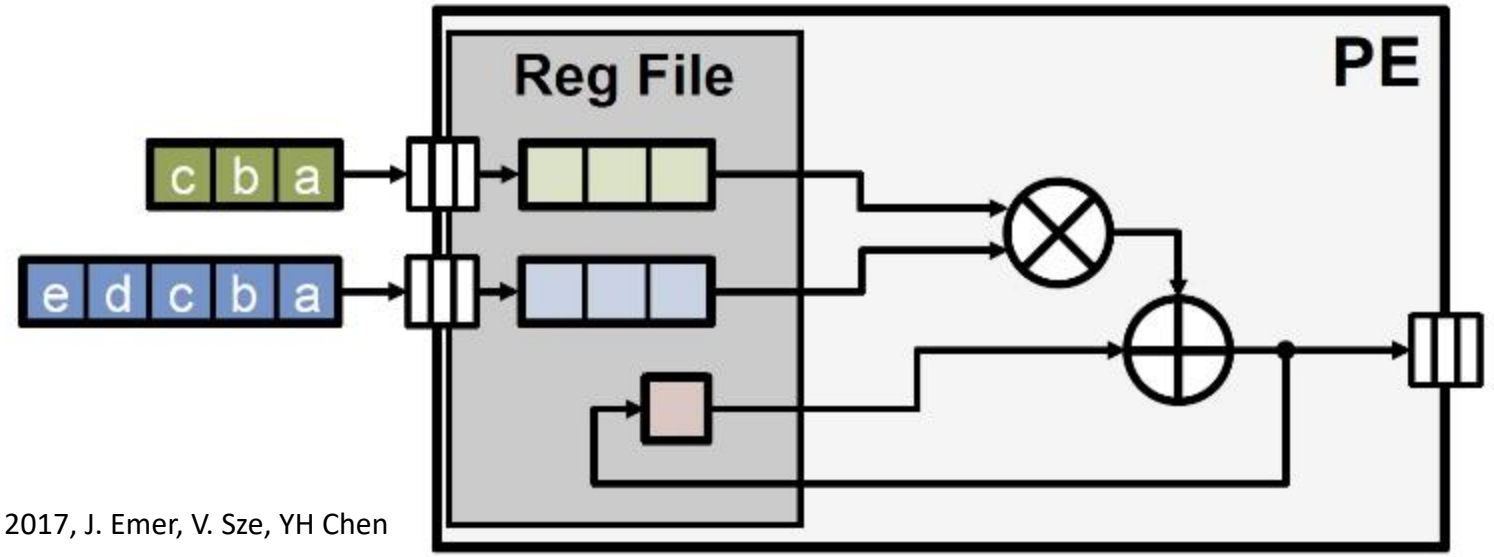
Fig. 5. Comparison of the architecture of original Eyeriss and Eyeriss v2. (a) Original Eyeriss. (b) Eyeriss v2.

Eyeriss v2: A Flexible Accelerator for Emerging Deep Neural Networks on Mobile Devices, Yu-Hsin Chen et al., IEEE JOURNAL ON EMERGING AND SELECTED TOPICS IN CIRCUITS AND SYSTEMS, VOL. 9, NO. 2, JUNE 2019

Enables Row Stationary data flow

- Hardware unrolling such that each PE gets a particular filter row / input row combination.
- RS+ extends this to include batch & channel dimensions for increased parallelism, but requires larger RF at each PE.
- Larger RFs, i.e. entire row ideally for RS, multiple rows time multiple channels for RS+.

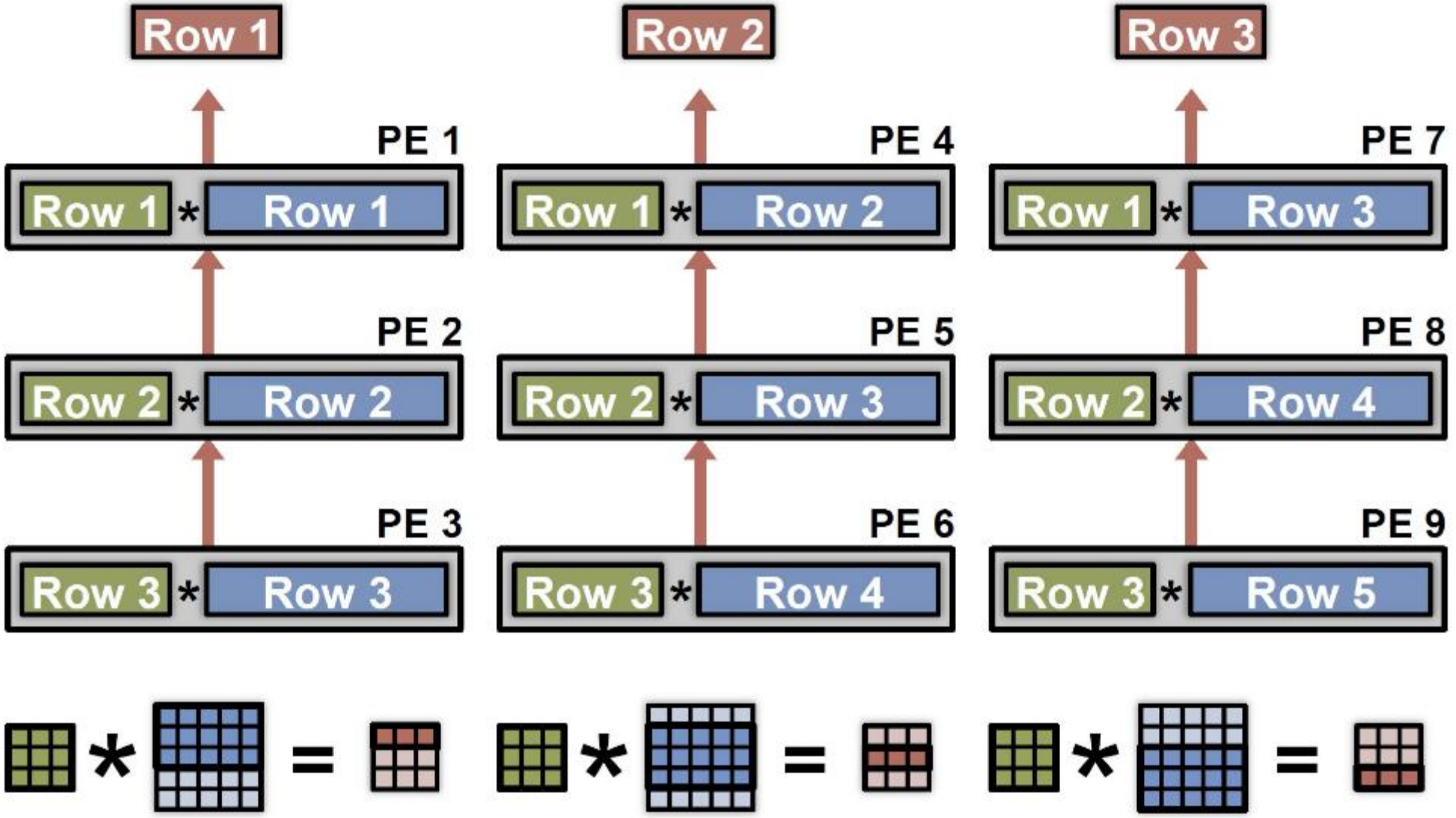




"DNN Accelerator Architectures", ISCA tutorial 2017, J. Emer, V. Sze, YH Chen

Input feature maps re-used diagonally

- Ifmap rows are re-used diagonally.
- But, requires large RF per PE.



"DNN Accelerator Architectures", ISCA tutorial 2017, J. Emer, V. Sze, YH Chen