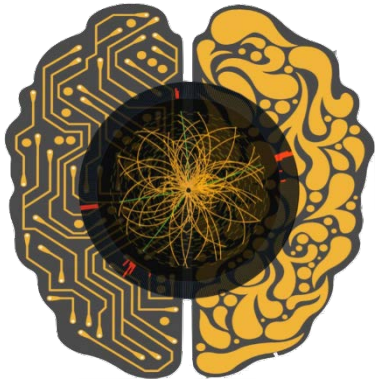


Real-time AI Systems
(Academia)

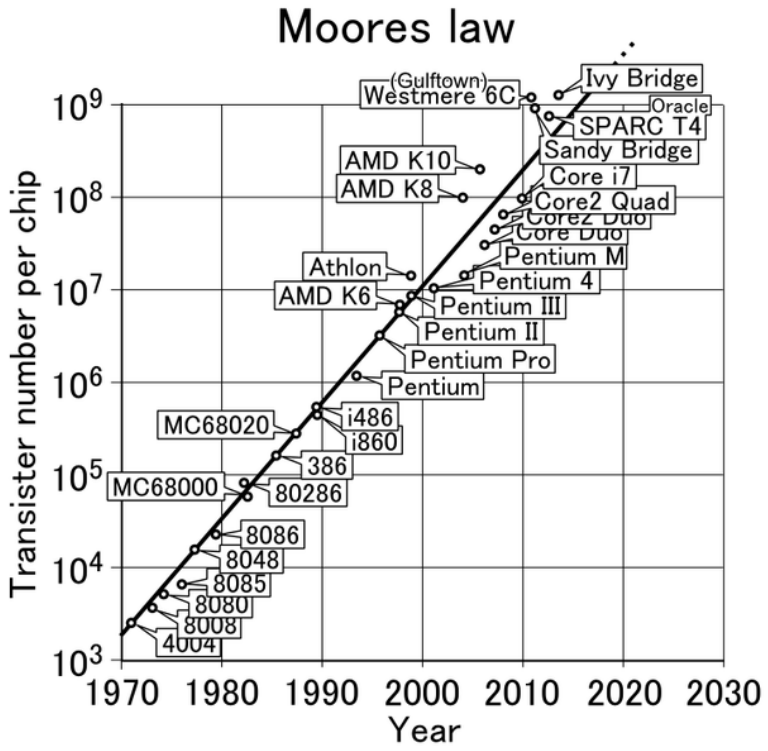
Giuseppe Di Guglielmo

giuseppe@cs.columbia.edu

Columbia University

Fast Machine Learning
Fermi Lab
September 10-13, 2019

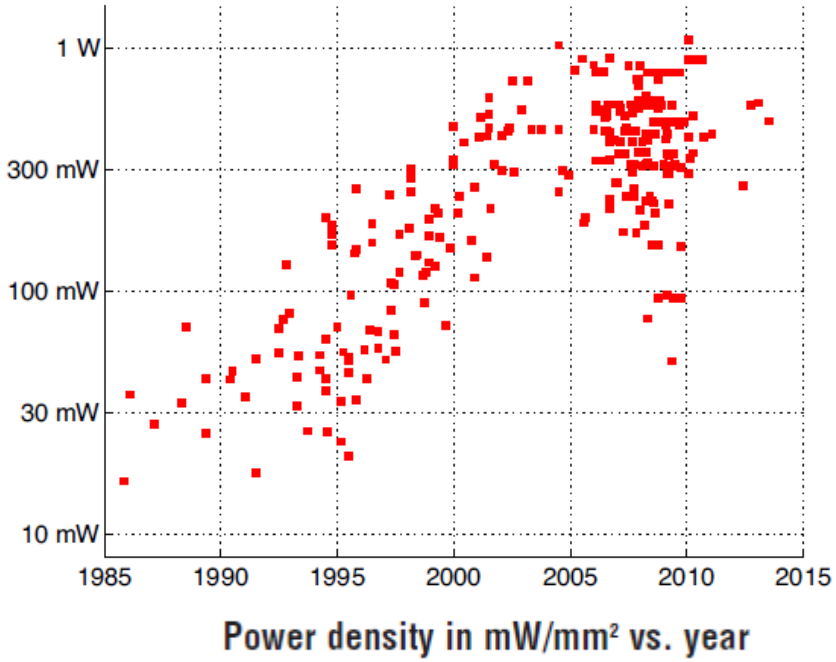
Technology Trends



- Moore's Law had many lives
  - 2004: The end of Moore's Law?
  - 2015: Beyond Moore's Law
  - 2016: After Moore's Law
  - 2017: A new way to extend Moore's Law

The Economist

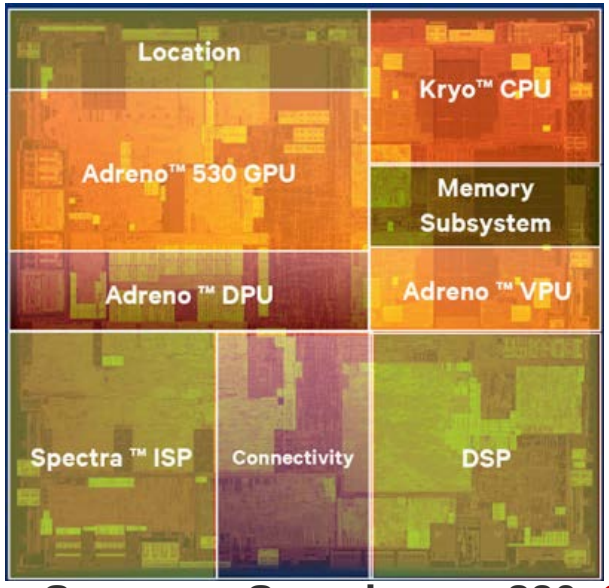
- … but Dennard's Scaling has stopped
  - On-chip power density cannot continue to grow



Source: Horowitz et al., 2014

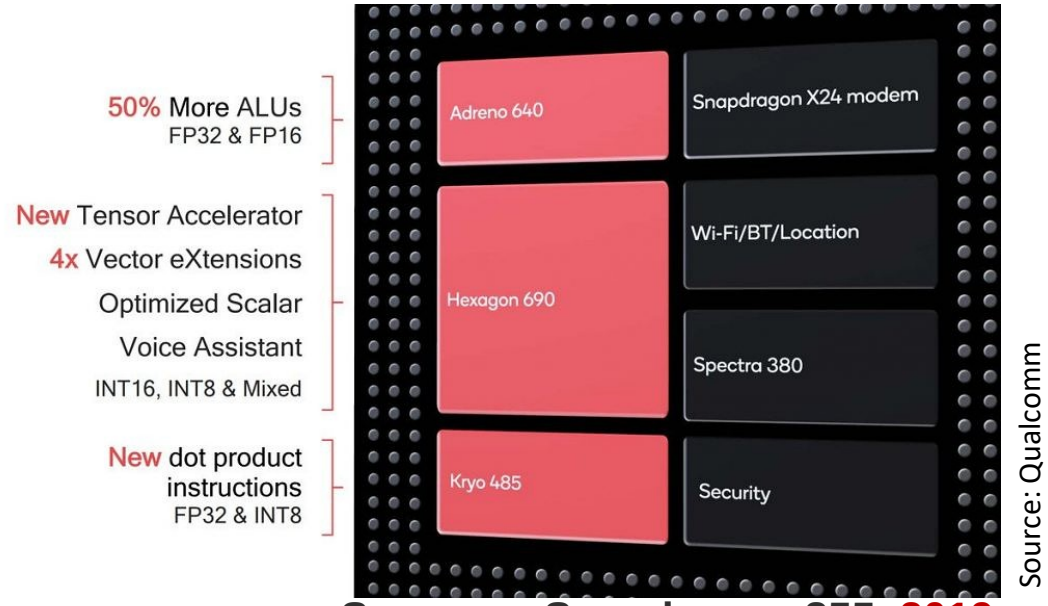
Emerging Computing Platforms for AI

- Heterogeneous Multi-Core Systems
  - Mix of processor cores and specialized hardware accelerators yields more energy-efficient computing
- The approach to AI is getting heterogeneous
  - DSP, GPU, CPU, FPGA, custom hardware…
    - … whatever is the best piece of hardware to handle an AI task in the most power efficient way



Source: Qualcomm

Samsung Snapdragon 820, 2016



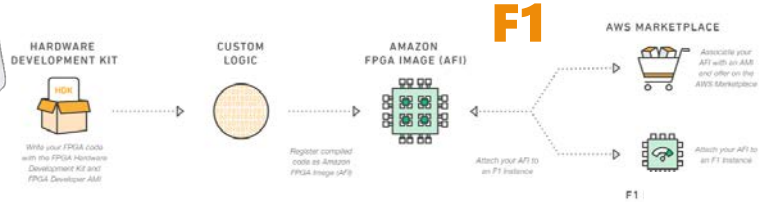
Source: Qualcomm

Samsung Snapdragon 855, 2019

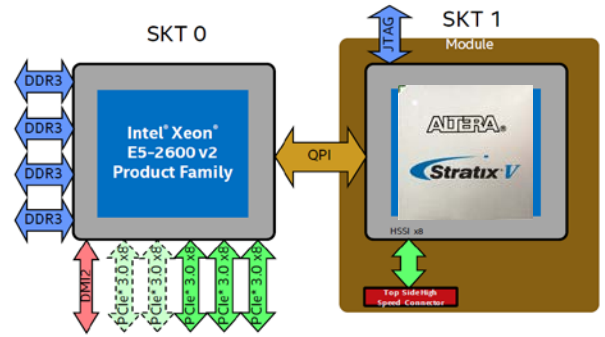
Heterogeneous Multi-core System-on-Chip

March 2017

May 2016 (1st gen.)
May 2017 (2nd gen.)

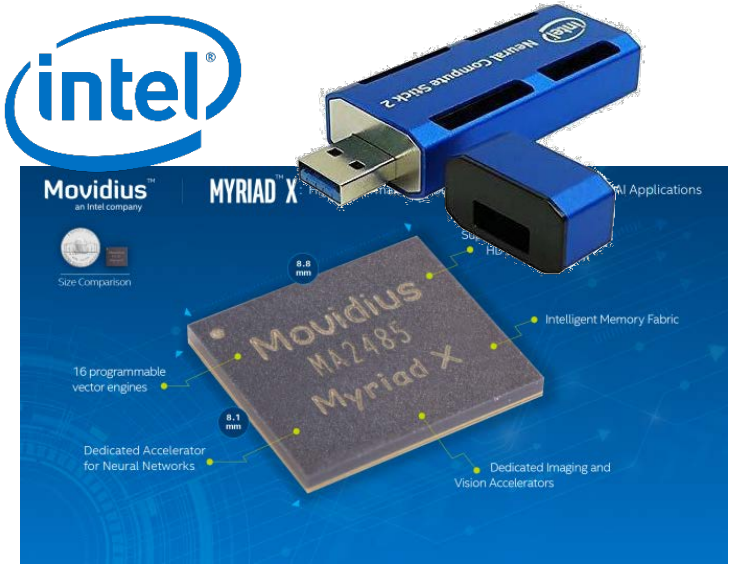


November 2016



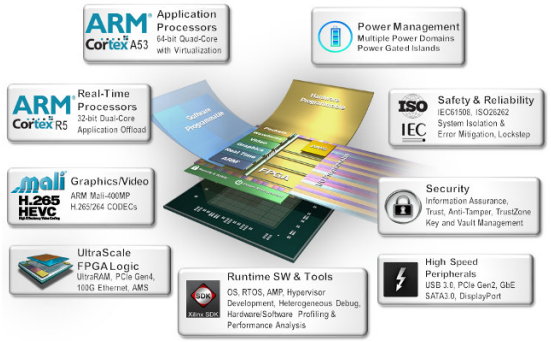
June 2016

Ivy Town Xeon + FPGA: The HARP Program

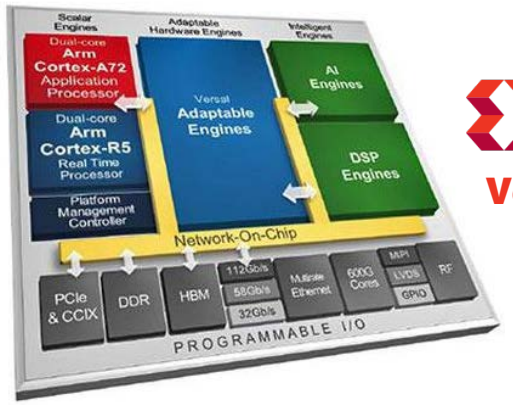


August 2017

Zynq UltraScale+ MPSoC



December 2013



XILINX

Versal

October 2018

Sources: Intel, Google, Xilinx, Amazon

Accelerator

- A special-purpose hardware that is optimized to perform a dedicated function(s) as part of a general-purpose computational system
  - While being part of a larger system, it spans a scale from being closely integrated with a general purpose processor to being attached to a computing system
  - Increasingly accelerators are migrating into the chip, thus leading to the rise of heterogeneous multi-core SoC architectures”
- Implemented as
  - Application Specific Integrated Circuit (ASIC)
  - Field-Programmable Gate Array (FPGA)

Source: CSEE_E4848, Columbia University

High-Level Synthesis

hls4ml

- RF
- Precision
- Latency/Resources

HLS Directives

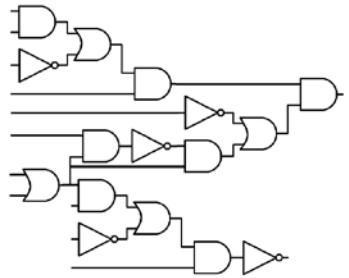
```
inline bar 'ON'
unroll l1 factor=10
bundle b1=A,B, b2=C
```

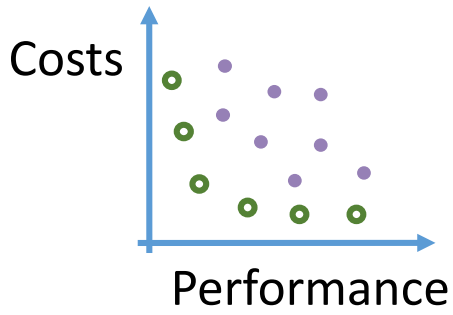
Software Specification

```
void foo(int A[10],
         int B[10], int C[10])
{
  int i = 0;
  l1: for (; i < 10; i++) {
    A[i] = B[i] * i;
  }
  i = 0;
  l2: for (; i < 10; i++) {
    B[i] = A[i] * B[i];
    C[i] = B[i] / 10;
  }
  bar(A, B);
}
```

HLS

Hardware Implementation(s)





AI > ML > NN

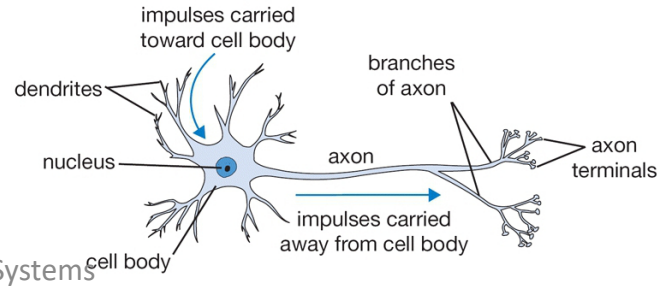
- Artificial Intelligence (AI)
  - Computer Vision
  - Pattern Recognition
  - …
  - Machine Learning (ML)
    - Linear Regression
    - K-Means Clustering
    - Decision Trees
    - …
    - Neural Networks (NN)
      - Convolutional Neural Networks
      - Binary Neural Networks
      - Recurrent Neural Networks
      - …

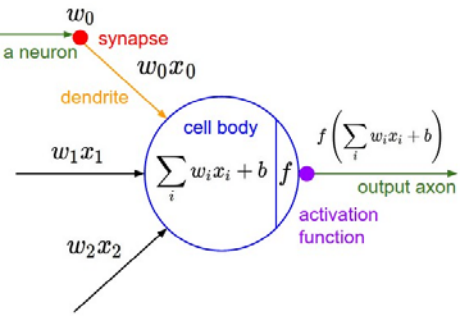
"Machine mimics cognitive functions such as learning and problem solving"

"Gives computers the ability to learn without being explicitly programmed"

{ Training Phase
  Inference Phase

"Mimics the physiology of human brain"

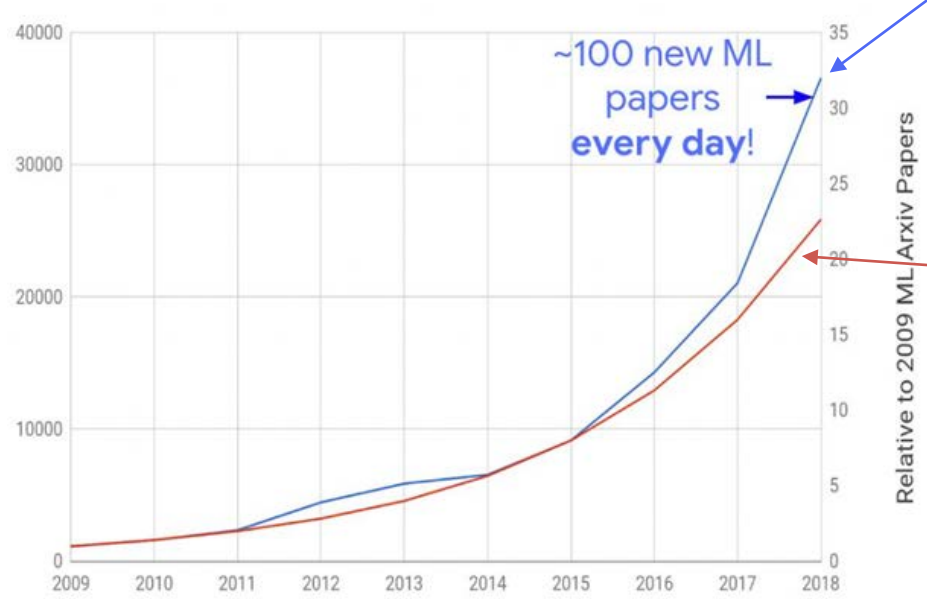




Source: CS231, Stanford University

Exponential Grow in Deep Learning

- There are several factors
  - Algorithms
  - Hardware
  - Data
  - Capital
  - Talent
  - Application

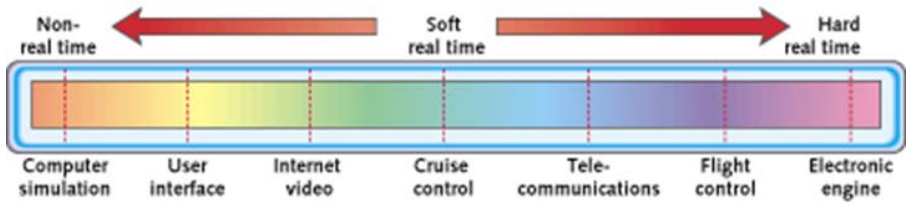


ML arXiv Papers

Moore's Law grow rate (~ 2x/2 year)

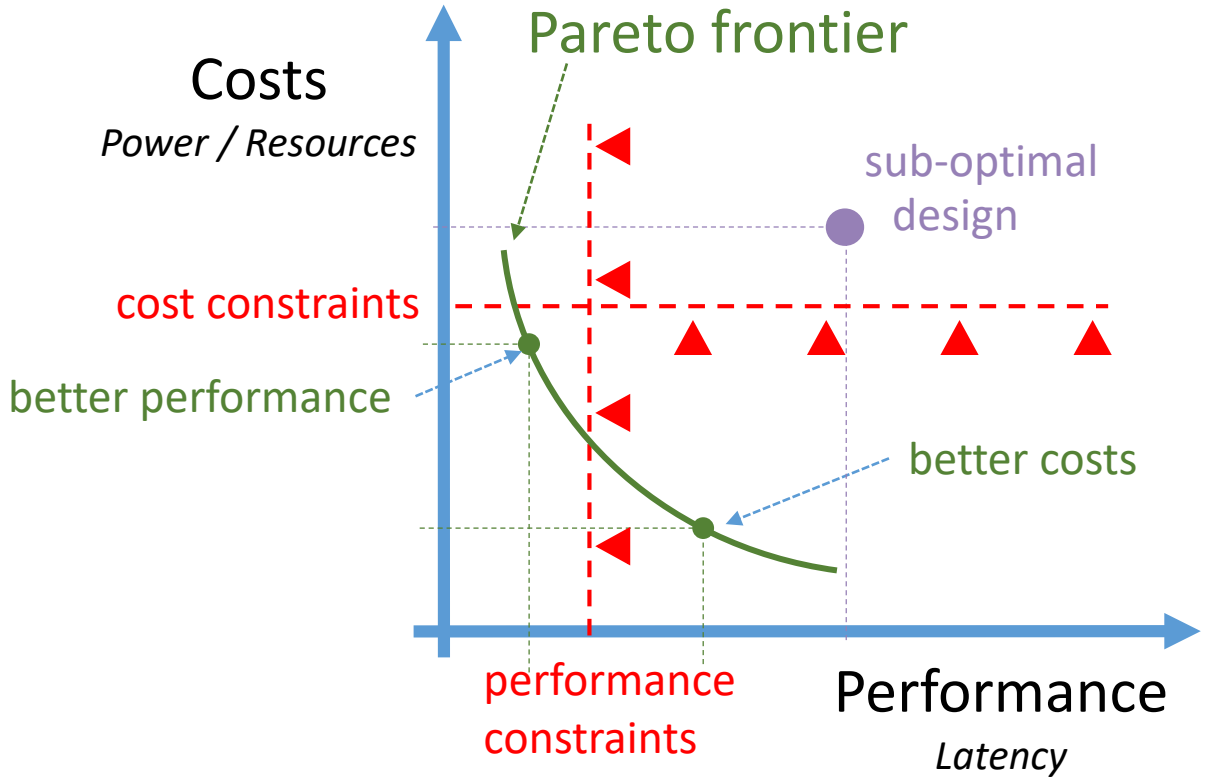
300,000x grow in computation between AlexNet and AlphaGo

(Hardware) Real-Time System



Source: eet.com

- Deadline-driven design
  - Constraints
    - Period/Frequency
    - Deadline
- Predicable system
  - "all tasks meet all deadlines"
- Deterministic system
  - A predictable system where the "timing behavior can be predetermined"



- HW design optimization space
  - Pareto optimality: "we cannot improve performance without paying costs, and vice versa"

A Too Short Blanket Problem

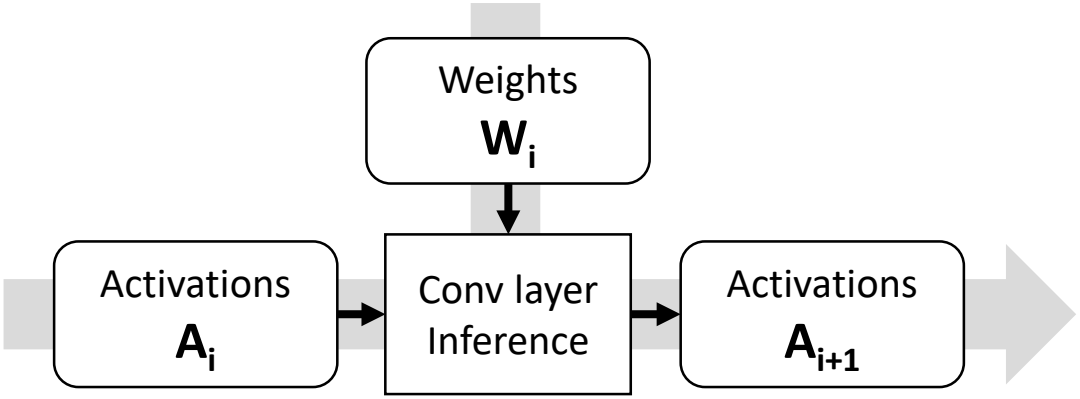
Source: http://dogatesketchbook.blogspot.com/2008/02/illustration-friday-blanket.html

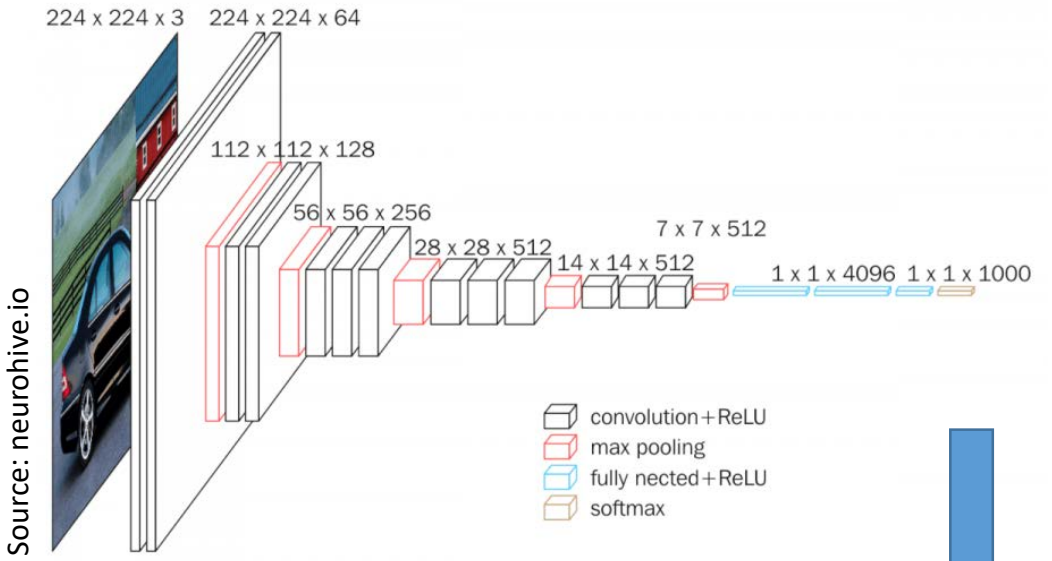
NN = Deterministic Performance

- We can statically compute
  - Memory occupation
  - Number of operations
  - Data movement
  - …

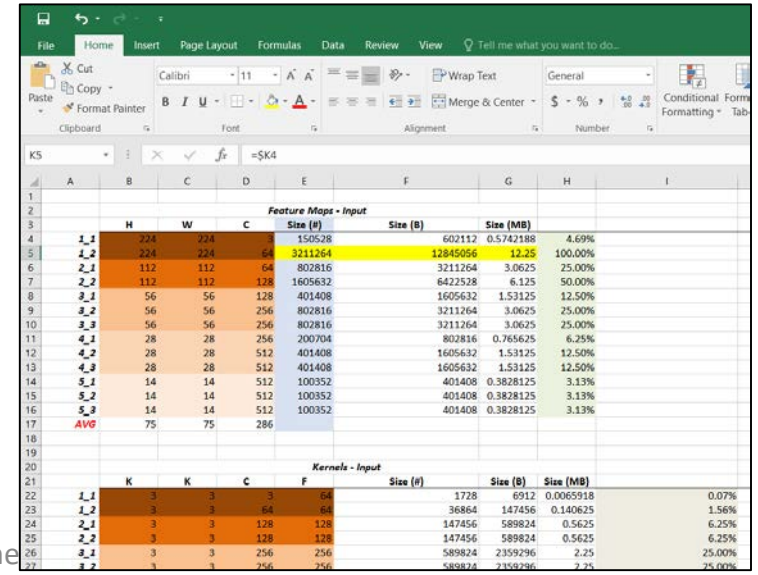
Convolutional layer

- $W_i = IN\_Ch_i * OUT\_Ch_i * F_i^2 * bits$
- $A_i = BATCH * F_i^2 * IN\_Ch_i * bits$
- $A_{i+1} = BATCH * F_i^2 * OUT\_Ch_i * bits$



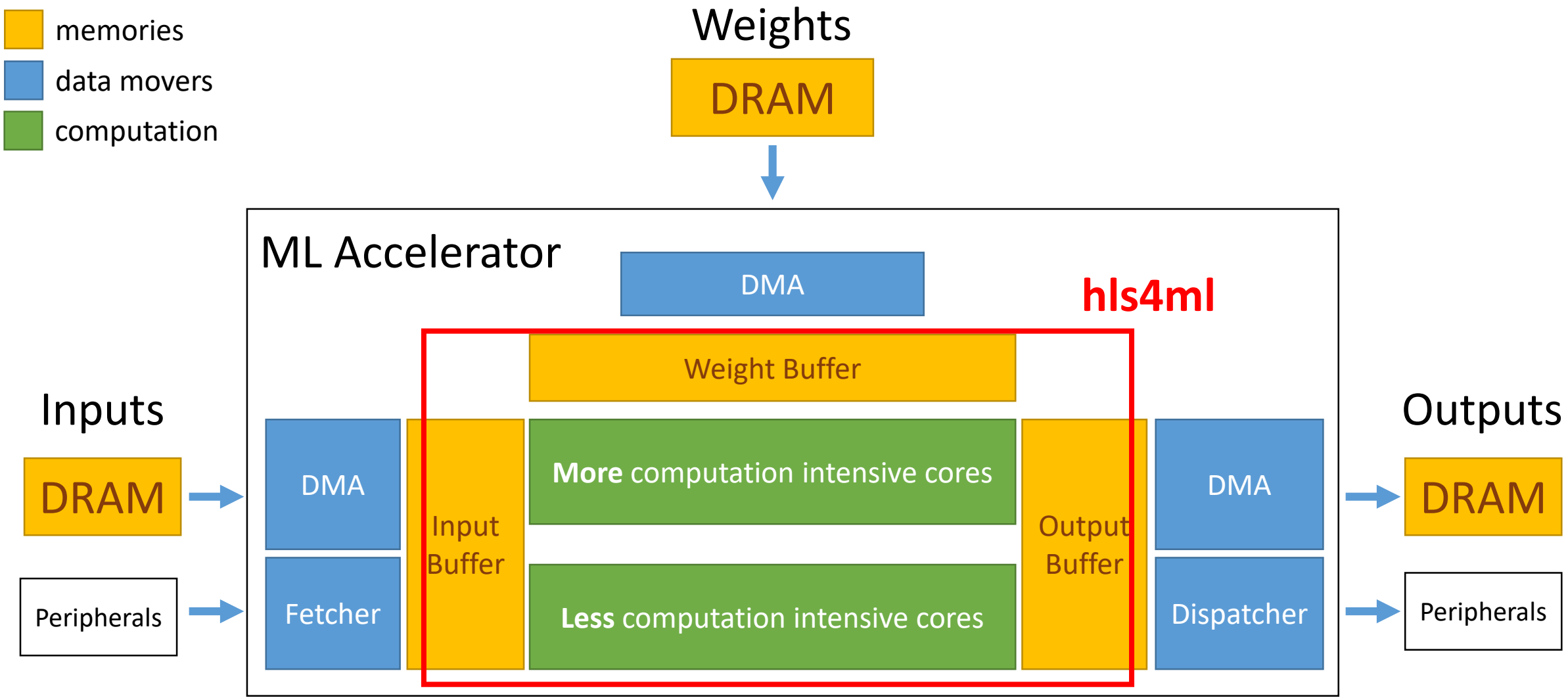


Source: neurohive.io



Big Excel file

Architecture of a ML Accelerator



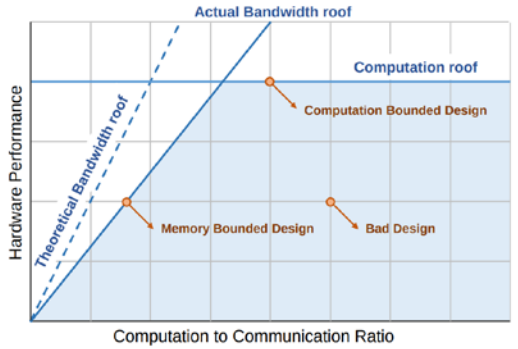
Pain Points*

memories

data movers

computation

Weights and Inputs fetching: bandwitdh vs. performance



A Survey of FPGA Based Neural Network Accelerator, K. Guo et al., May 2018

Weights

Caching and interconnects may introduce unpredictability

DRAM

Power comsumption (embedded domain)

Latency (real-time domain)

ML Accelerator

DMA

Weight Buffer

Huge models that spill into DRAM

Inputs

DRAM

Peripherals

DMA

Fetcher

Input Buffer

More computation intensive cores

Less co…e cores

Huge amount of computation (hard to design)

Output Buffer

DMA

Dispatcher

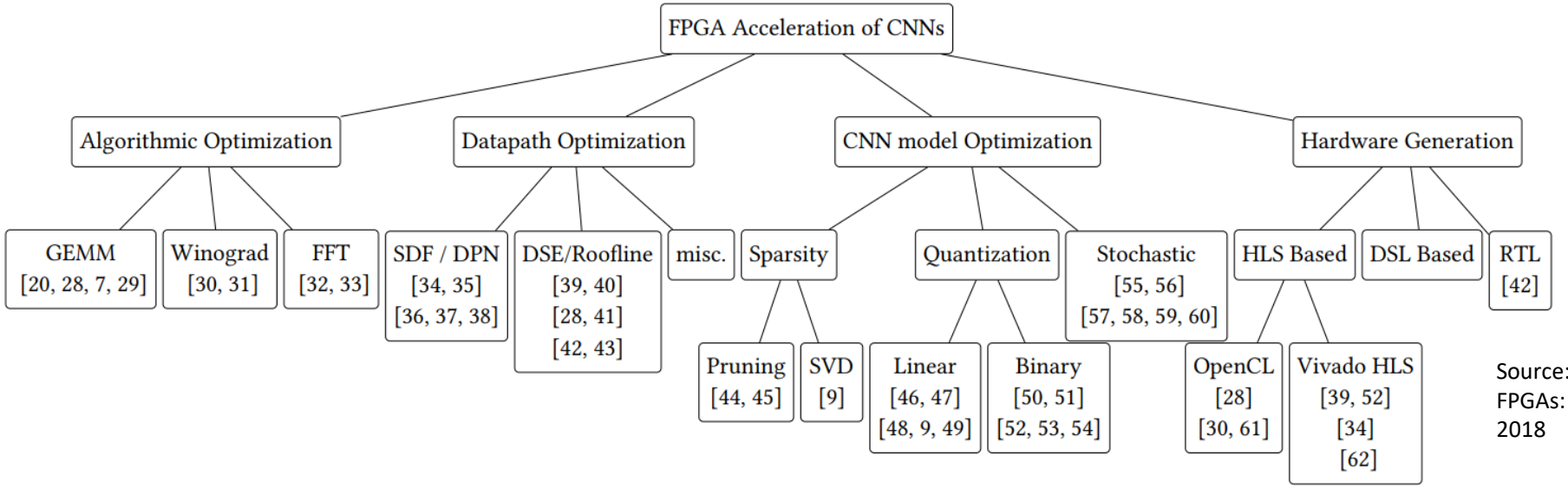
Outputs

DRAM

Peripherals

* Michaela Blott, Principal Engineer in Xilinx, Architecture for Accelerating DNNs, Hot Chips 2018

Cost & Performance Optimization Techniques



Source: Accelerating CNN Inference on FPGAs: A Survey, K. Abdelouahab et al., 2018

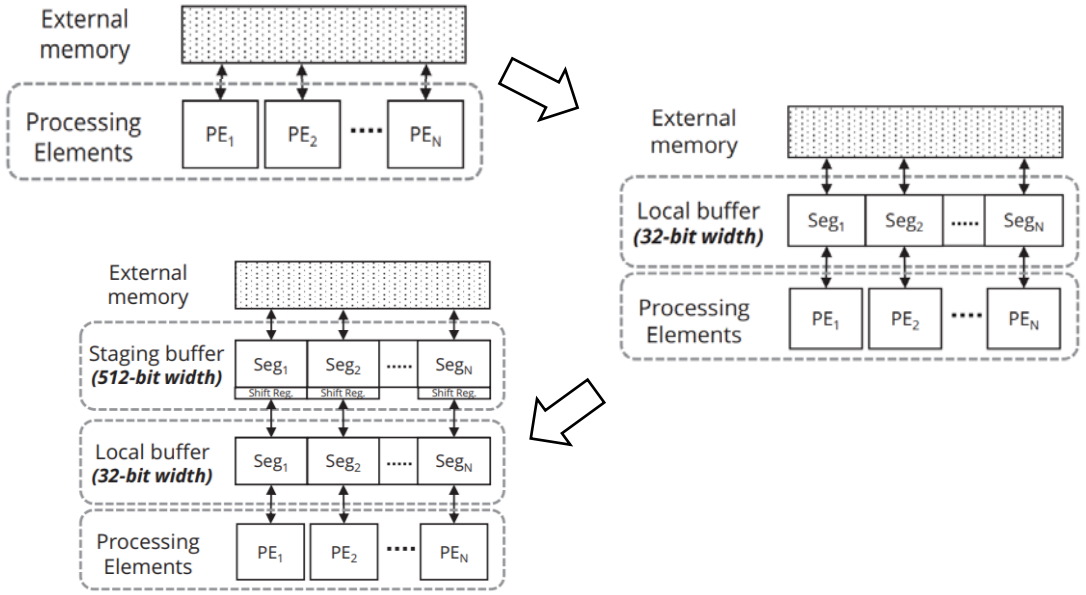
- Accelerating CNN Inference on FPGAs: A Survey, K. Abdelouahab et al., Jan 2018
  - https://arxiv.org/pdf/1806.01683
- Toolflows for Mapping Convolutional Neural Networks on FPGAs: A Survey and Future Directions, S. I. Venieris et al., Mar 2018
  - https://arxiv.org/pdf/1803.05900
- A Survey of FPGA Based Neural Network Accelerator, K. Guo et al., May 2018
  - https://arxiv.org/pdf/1712.08934
- Optimizing Memory Efficiency for Deep Convolutional Neural Networks on GPUs, C. Li et al, 2016
  - https://arxiv.org/pdf/1610.03618

A Good Start is Half of the Work

- Kernel (Task)
  - Functionally and computationally important portion of an application
  - Well defined interface (inputs and outputs)
  - E.g. Conv2D
- Algorithm
  - An algorithm solves a particular kernel
  - E.g. Conv2D via GEMM, Winograd, FFT
- Implementation
  - Different implementations may exist of the same algorithm
  - HLS is very sensitive on the coding style

Source: MachSuite: Benchmarks for Accelerator Design and Customized Architectures, David Brooks, 2014

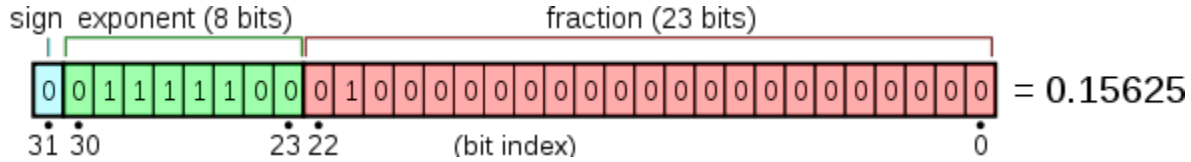
- Loop transformations to minimize memory access
  - Memory layout
- Create local memory buffers and keep data on-chip as much as

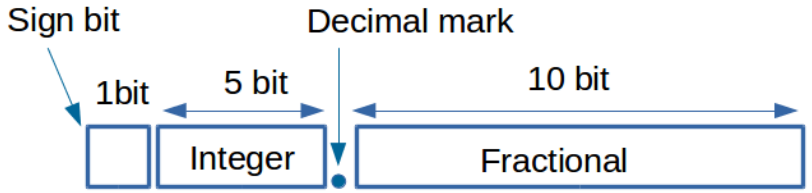


Source: Bandwidth Optimization Through On-Chip Memory Restructuring for HLS, Jason Cong et al., 2017

Reducing Bit-Precision

- Arithmetic
  - Floating Point
    - FP64, FP32, FP16, FP11 …
  - Fixed Point (Linear Quantization)
    - ✓ A lot of freedom
    - ✓ "Essentially" integer arithmetic
    - ✗ Overflow and underflow problems
  - Binary quantization
- Linear reduction in memory footprint
  - Reduce the amount of data transfer
  - Model may fit local buffers (on-chip)
- Reduction of the arithmetic logic
  - Improve area, power, latency

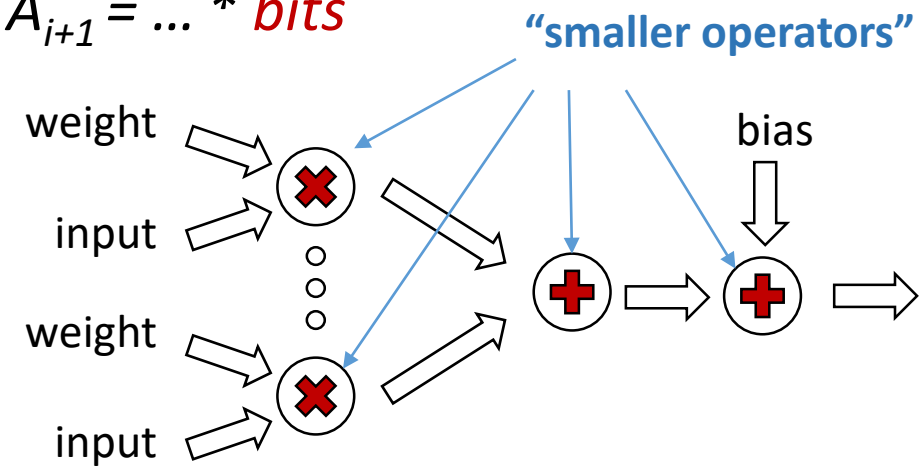




W_i = … * *bits*

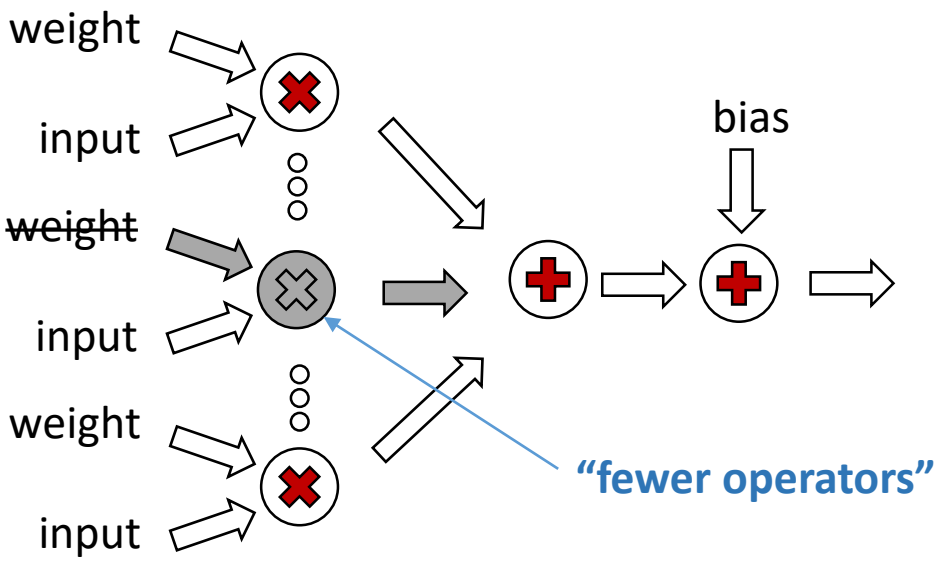
A_i = … * *bits*

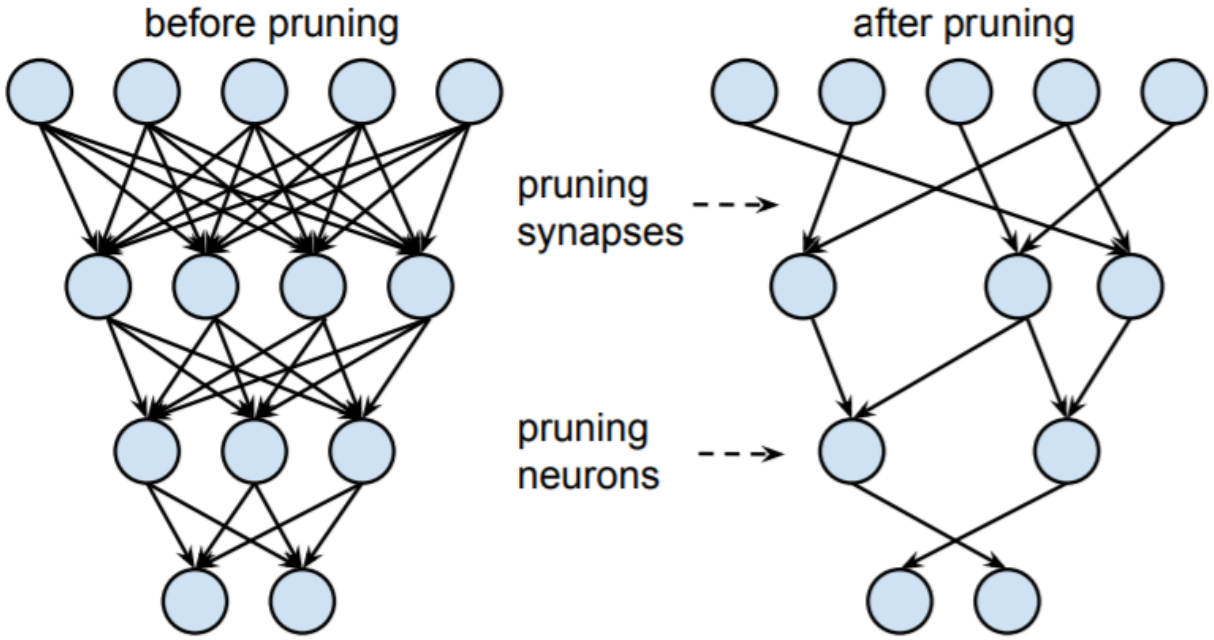
A_{i+1} = … * *bits*



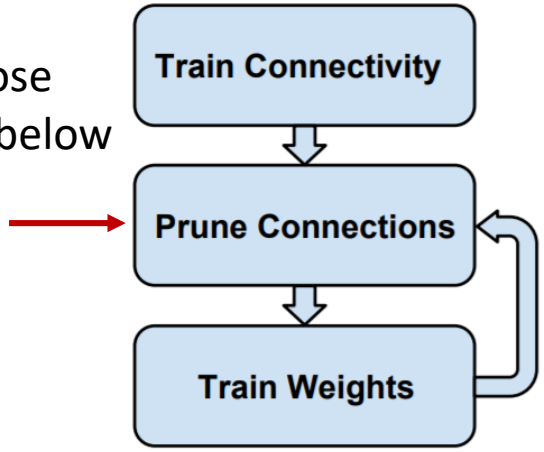
Pruning

- Goal: Reduce storage, data transfer, and computation
- Reduction of the model size without loss of prediction accuracy
  - Alexnet 9x, VGG-16 13x





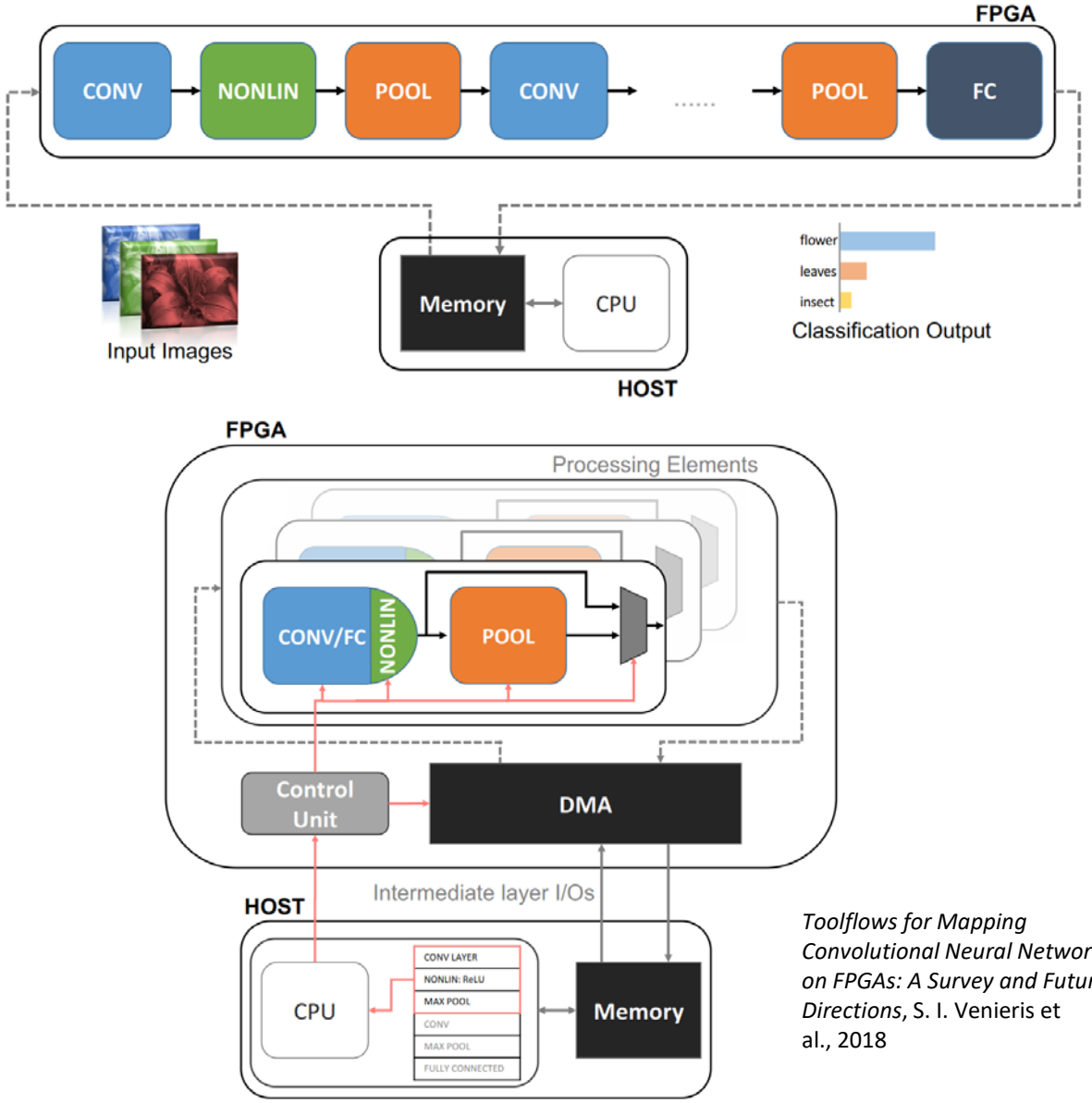
"We throw away those edges with weights below a certain threshold"



Source: Learning both Weights and Connections for Efficient Neural Networks, Song Han, 2015

Hardware Architecture

- Streaming
  - One distinct hardware block for each CNN layer, where each block is optimized separately to exploit the inner parallelism of the layer
    - Weights: on-chip or off-chip
    - Controller: software or hardware
- Single Computation Engine
  - Single computation engine that executes the CNN layers sequentially
    - Software controller of the scheduling of operations (may be inefficient)
    - Flexibility and reusability over customization
    - “One-size-fits-all” approach has higher performance on NN with a uniform structure



Toolflows for Mapping Convolutional Neural Networks on FPGAs: A Survey and Future Directions, S. I. Venieris et al., 2018

Data Layout in Main Memory

- CNNs work on 4-dimensional matrices
  - Data can be stored in memory in 24 (=4!) different ways
- Data layout determines the memory-access patterns and has critical performance impact
- Algorithms and data layout should be designed to minimize the number of DMA transactions
  - Rule of thumb:
    - «Fewer and longer transactions»

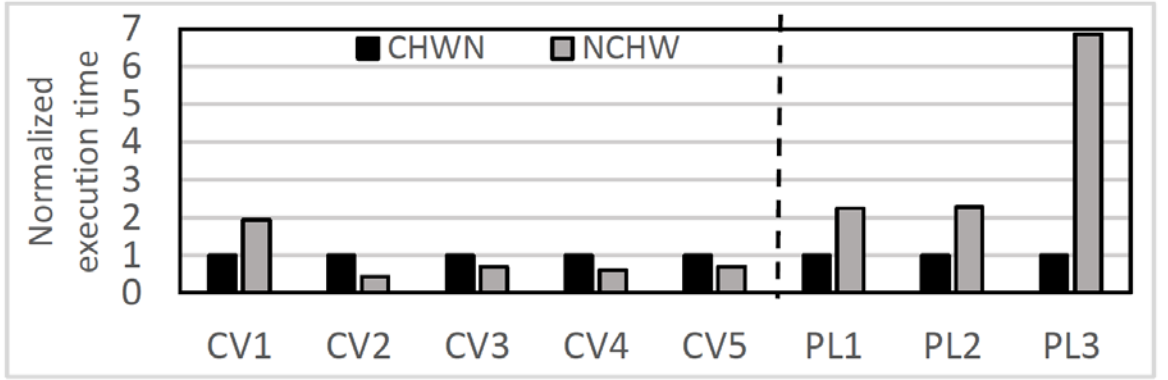
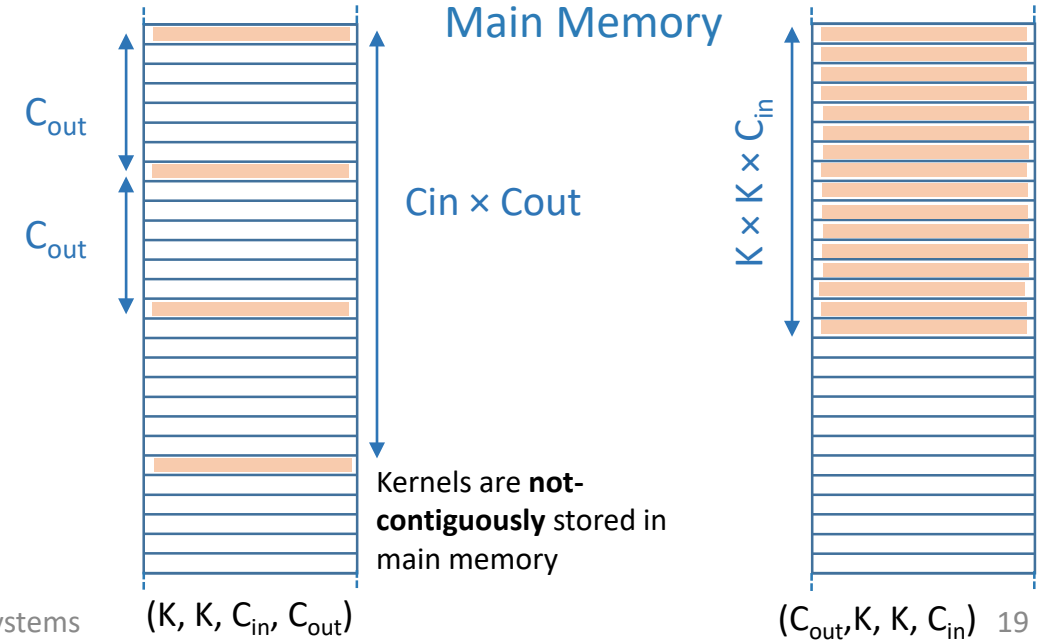


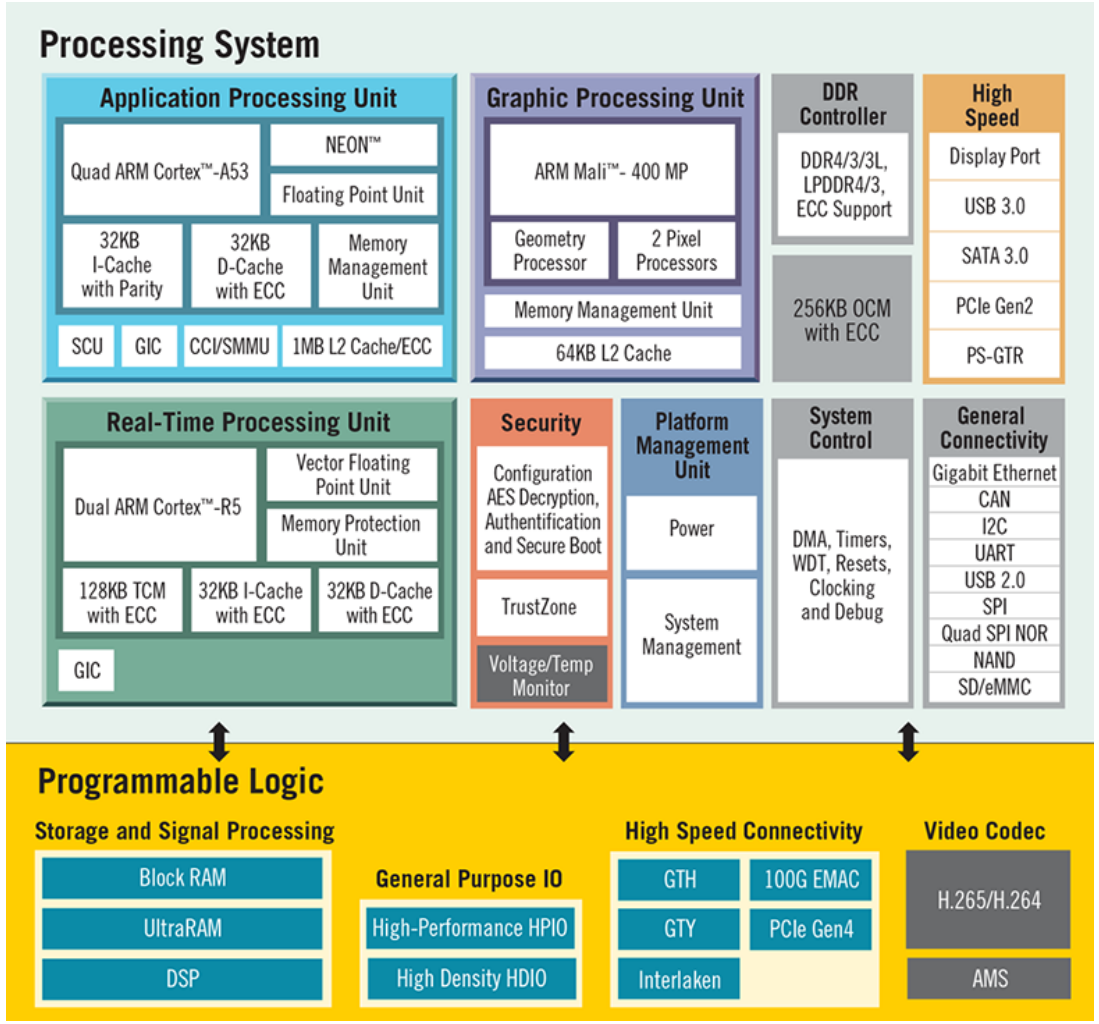
Fig. 1. Performance comparison between the CHWN layout (cuda-convnet2) and NCHW layout (cuDNNv4) on convolutional and pooling layers in AlexNet [12]

Optimizing Memory Efficiency for Deep Convolutional Neural Networks on GPUs, C. Li et al, 2016



OS, Memory Hierarchy, RT-CPU

- Real-time operating system
  - Predictable / Deterministic
  - Concept of running tasks
    - Read in data over an interface
    - Perform an operation on the data
    - …
- “Caches are evil”
  - Disable caches
  - Cache partitioning



Conclusions

- The design of a real-time AI system is more complicated than just meeting all of the deadlines
- Trade off between your timing constraints and area and power costs
- Exciting area to work on because of the constantly increasing importance of AI and variety of application domains

Source: http://dogatesketchbook.blogspot.com/2008/02/illustration-friday-blanket.html

Q/A

Thank you!