

Why Xilinx for Machine Learning?



Craig Abramson
Senior Technical Marketing Engineer
Longmont, CO

CELEBRATING 15 YEARS!

Electronic

White Paper: AI Engines

ut



WP506 (v1.0.2) October 3, 2018

Xilinx AI Engines and Their Applications

Subscribe now



feature article



For compute-intensive machine learning DNN/C AI Engines are an architecture that delivers 50% the power consumption of logic solutions.

Versal ACAP AI Core Series Product Selection Guide

This white paper explores Xilinx's new AI Engine for machine learning DNN/C 5G requires between five with prior generations; AI the throughput and comp accelerated speed require The emergence of machine networks, dramatically inc AI Engines, which are opt density to meet these der by as much as 50% when programmable logic. AI Engines are programm programmers. AI Engines Engines to provide a high



privately held, ma algorithms and sy investment in Dec (Cue the old TV a shaver company,

According to Xilin

"...the two compa pruning technolo

Tech's neural network

© Copyright 2018 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. AMBA, AMBA Designer, ARM, ARM1176JZ-S, CoreSight, Cortex, and PrimeCell are trademarks of ARM in the EU and other countries. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. All other trademarks are the property of their respective owners.

WP506 (v1.0.2) October 3, 2018

www.xilinx.com

1



Machine Learning Challenges



The rate of AI innovation



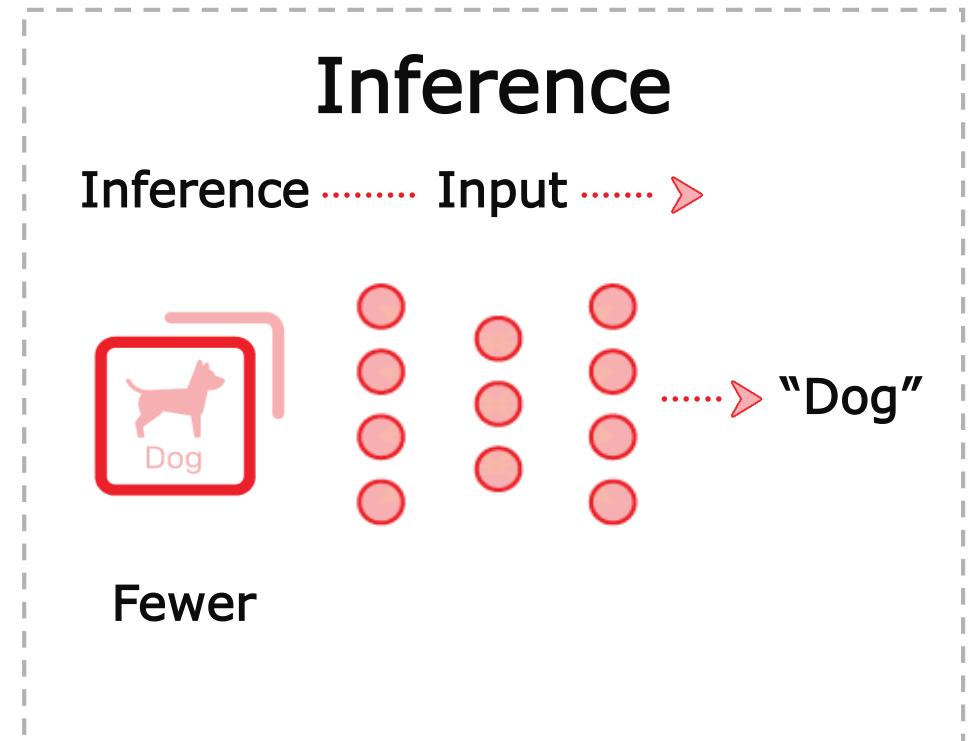
Performance at low latency



Low power consumption



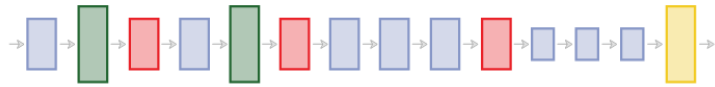
Whole app acceleration



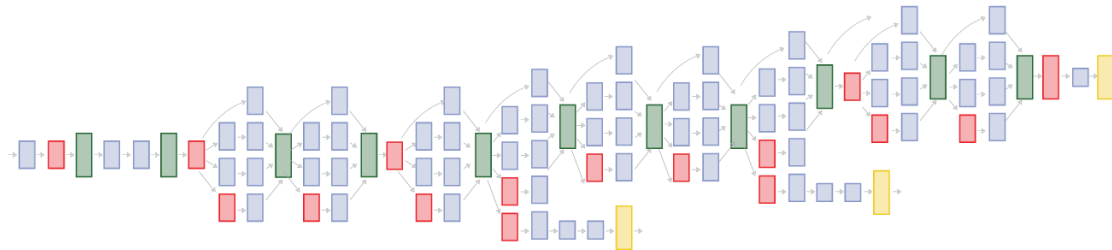
Rate of Innovation Outpaces Silicon Cycles



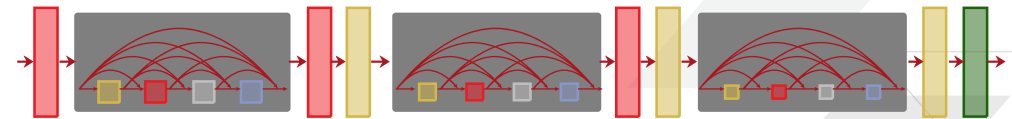
AlexNet



GoogLeNet



DenseNet



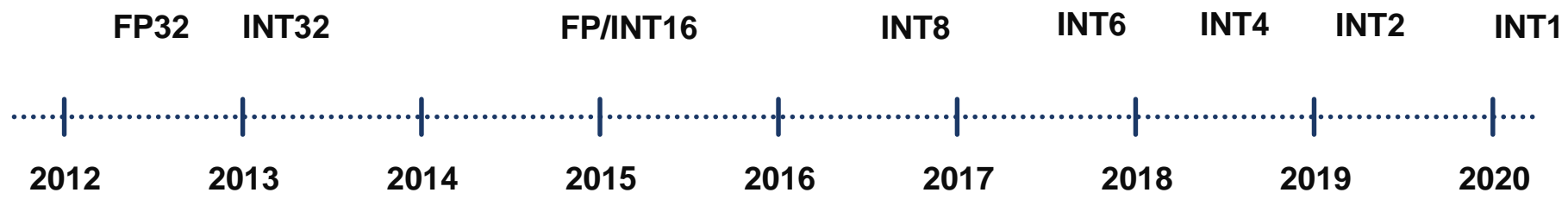
2012

2018

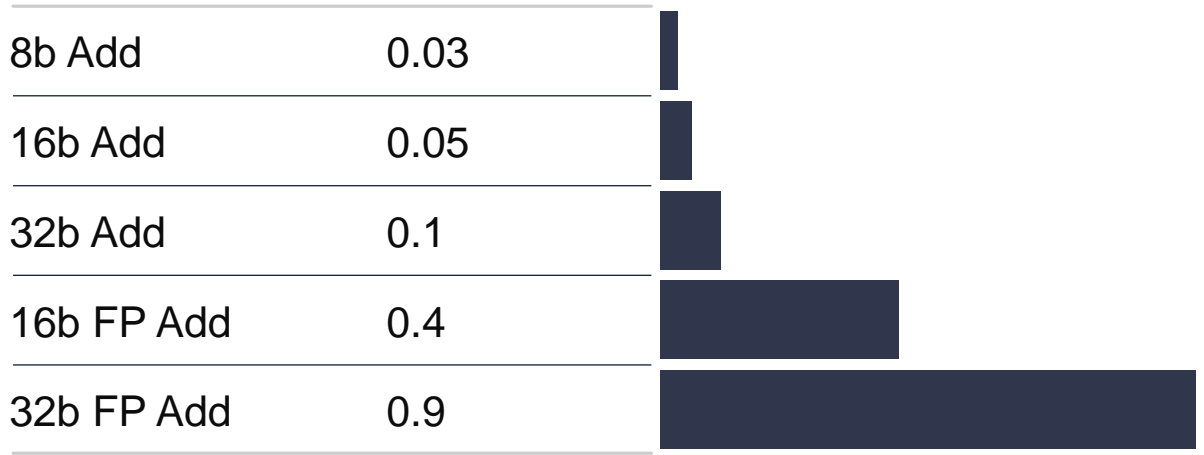
Silicon lifecycle



Inference is Moving to Lower Precision

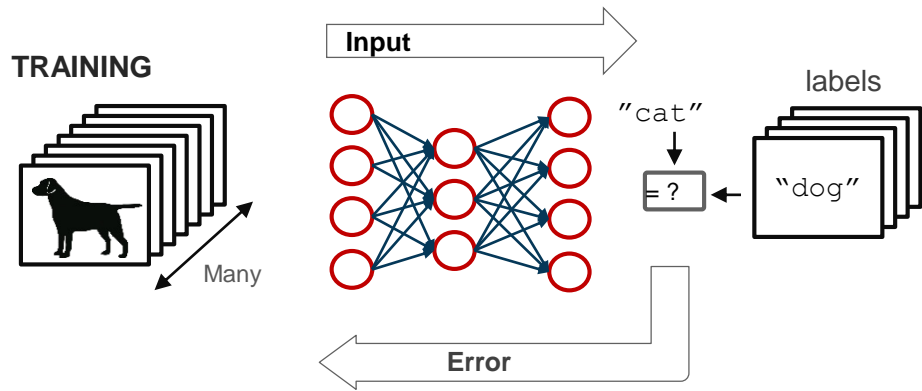


RELATIVE ENERGY COST

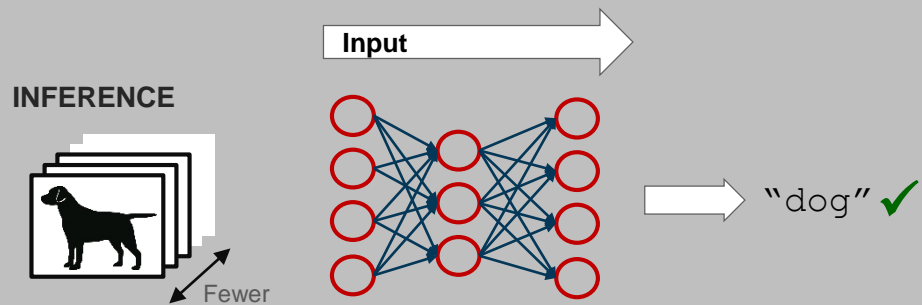


Source: Bill Dally (Stanford), Cadence Embedded Neural Network Summit, February 1, 2017

Machine Learning Inference is Xilinx Focus



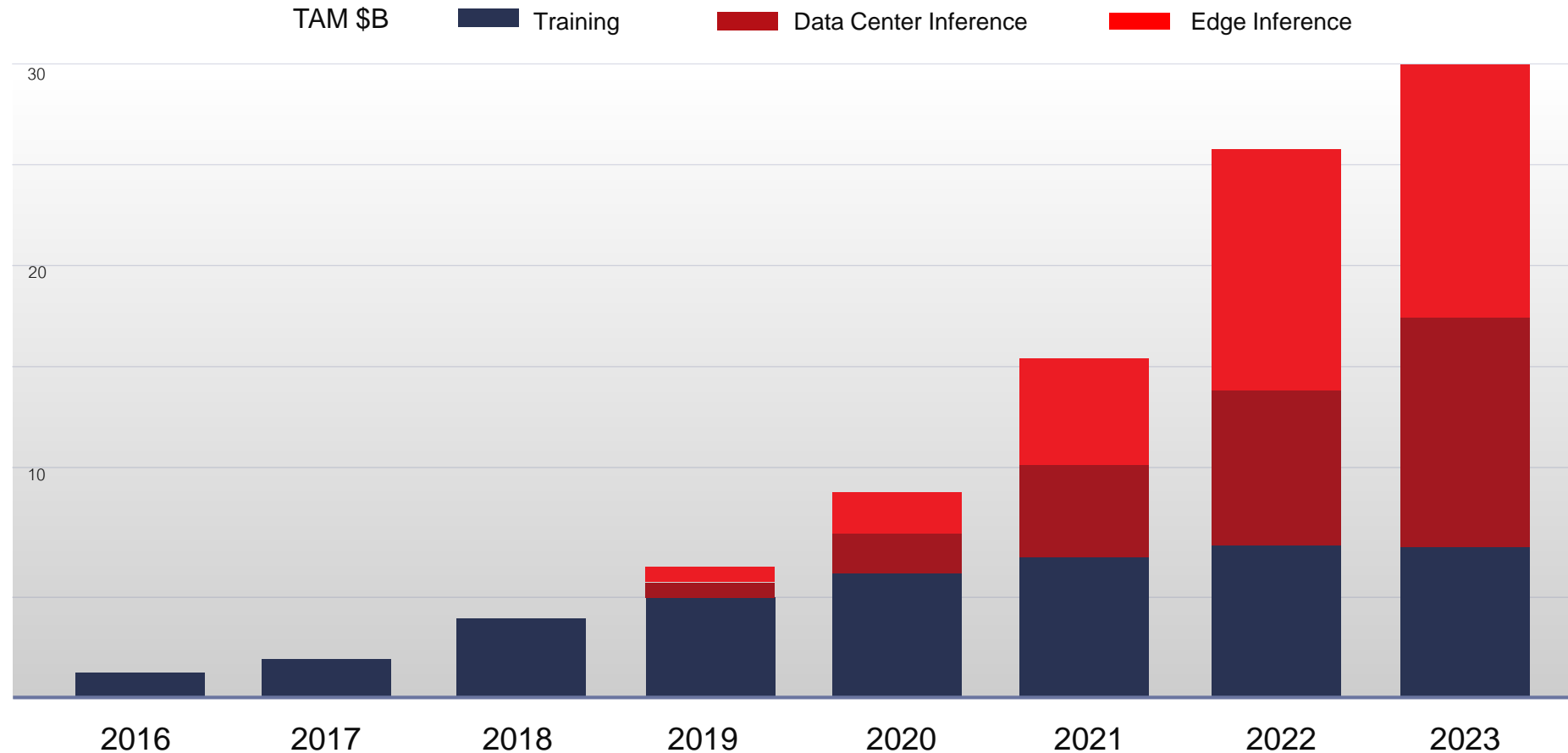
Training: Process for machine to "learn" and optimize model from data



Focus


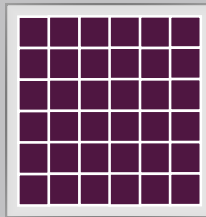
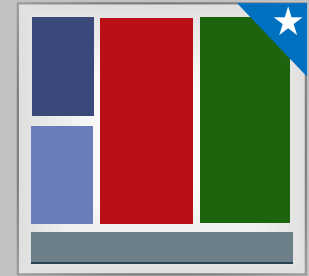

Inference: Using trained models to predict/estimate outcomes from new observations in efficient deployments

Why ML Inference? It's Where the Market is going to be...



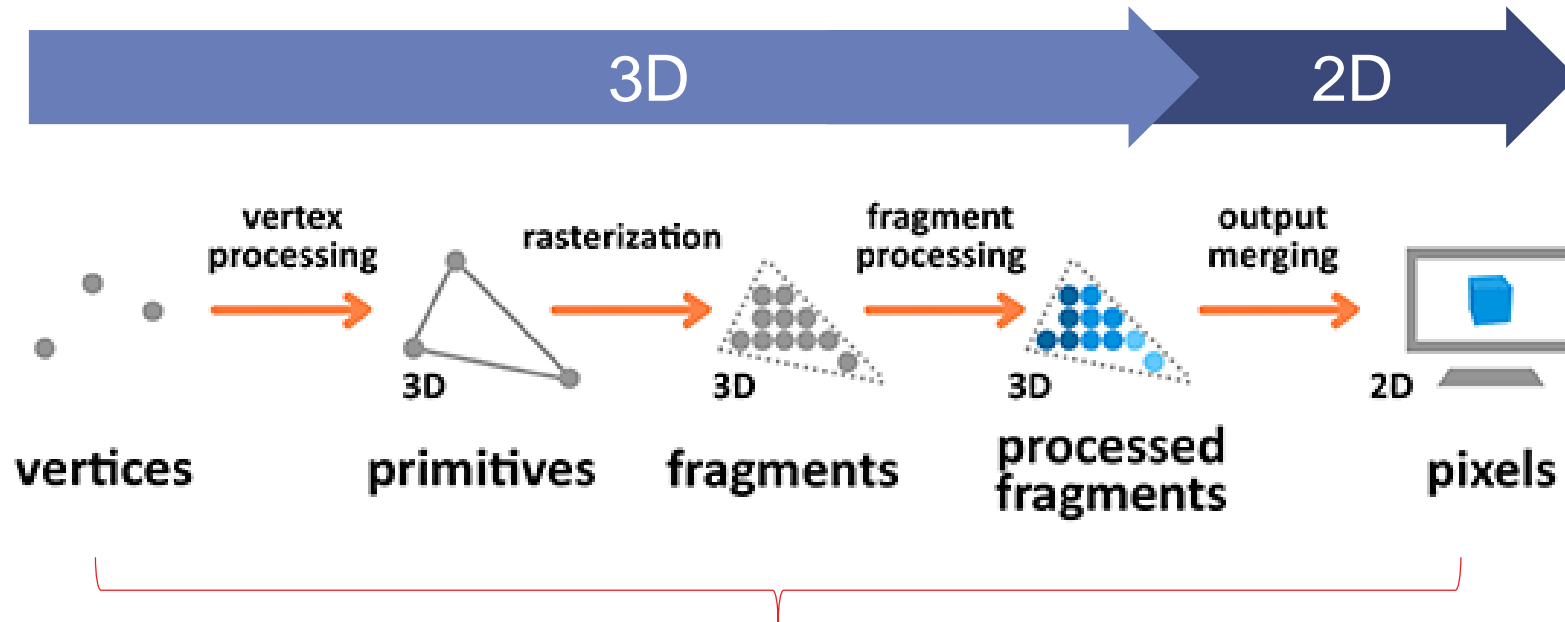
Barclays Research, Company Reports May 2018

Delivering Adaptable ML Compute Acceleration

	CPU (Sequential) 	GPU (Parallel) 	FPGA / SoC 	Custom ASIC 
SW Programmable	✓	✓	✓	✓
HW Adaptable	—	—	✓	—
Workload Flexibility	✓	✓	✓	—
Throughput vs. Latency	—	—	✓	✓
Device / Power Efficiency	—	—	✓	✓

Why GPUs in the First Place?

GPU Graphics Pipeline: Converts 3D representations of images into 2D space



Extensive Matrix
Math & Manipulation

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}_A \times \begin{bmatrix} e & f \\ g & h \end{bmatrix}_B = \begin{bmatrix} ae + bg & af + bh \\ ce + dg & cf + dh \end{bmatrix}_C$$

Machine Learning Requires
Similar Operations

- Matrix Convolution
- Matrix Multiplication

“Can we apply GPUs to other problems?”

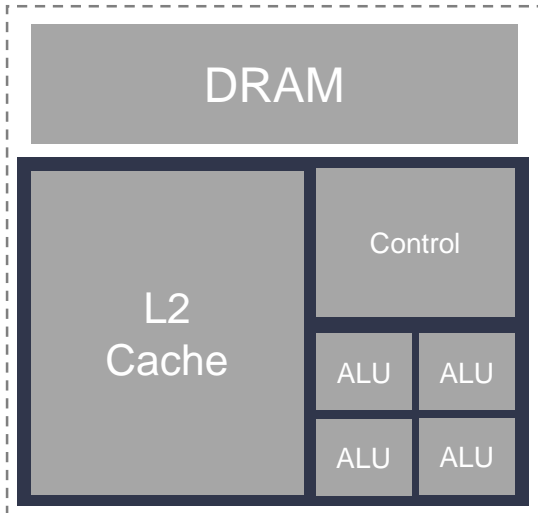
CPU / GPU Architecture

Low Latency
is a
Relative Term

CPU

Latency Optimized

- Serial Processing
- Large Data Cache



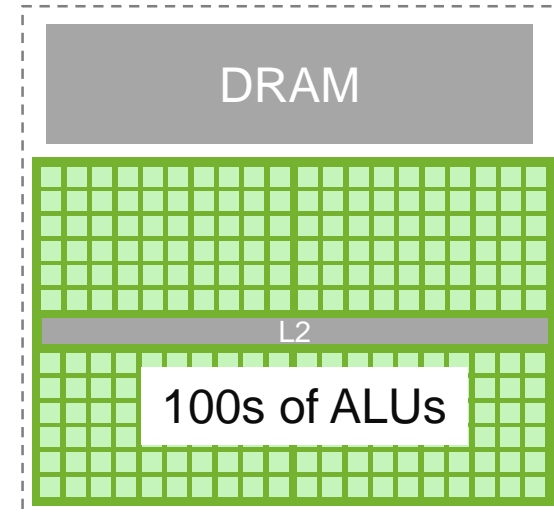
Cache Access



GPU

Throughput Optimized

- Data Parallel Processing
- Compute favored over on-chip Memory

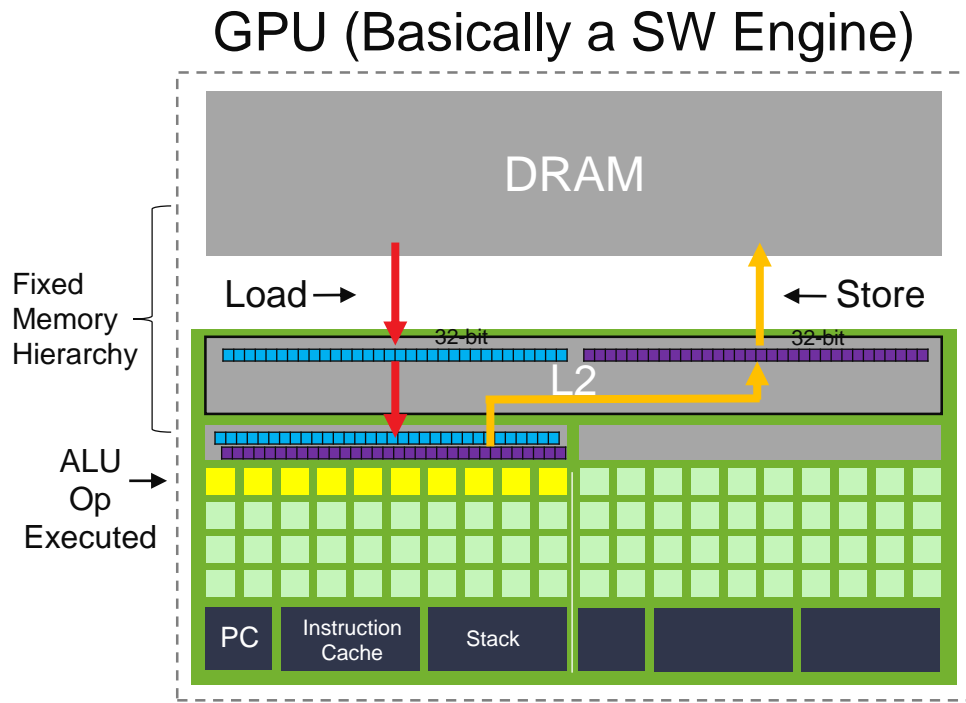


Memory Access



“Latency Hiding”
100’s of Threads
GPUs “BATCH”
data to get
around this.
More later . . .

Data Flow and Data Precision Matters

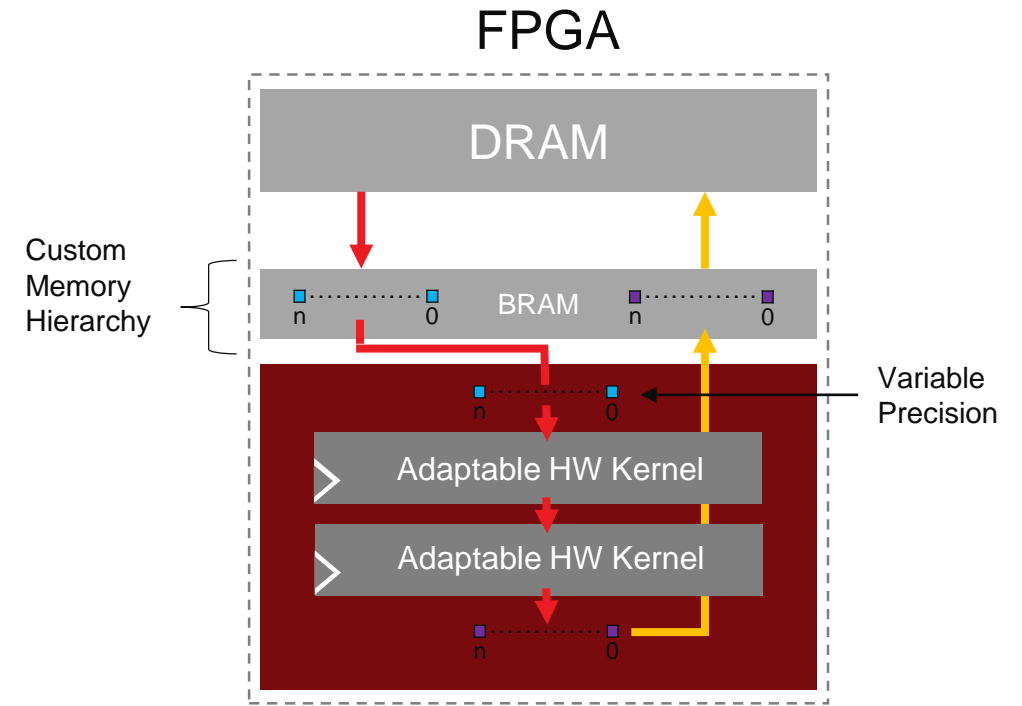


> Software Defined Data Flow

- >> Major overhead (memory, comms, power)
- >> Non-deterministic Behavior (latency)

> Fixed Data Precision Support

- >> Floating point / Integer units
- >> Native precisions defined at T/O



> Hardware Defined Data Flow

- >> Minimum overhead, custom compute / memory
- >> Deterministic Behavior (latency)
- >> Reconfigurable to current / future workloads

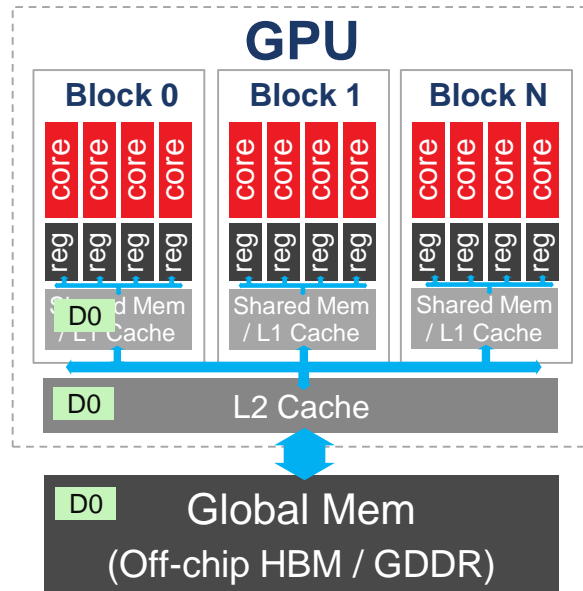
> Variable Data Precision Support

- >> Optimize for [memory, power, cost](#)
- >> Future proof, adapts to future precision trends

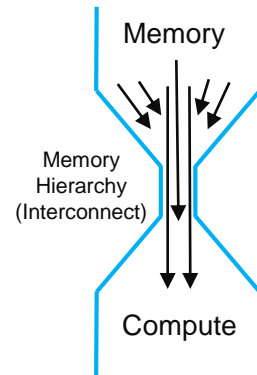
Memory Hierarchy: Very Fundamental FPGA Advantage

Latency : Power

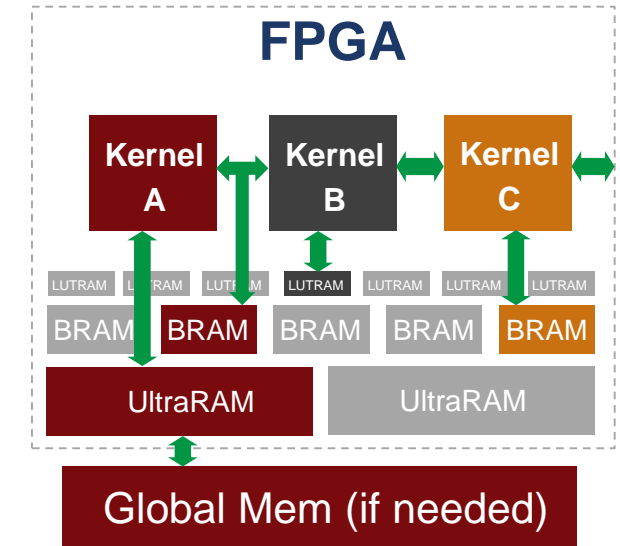
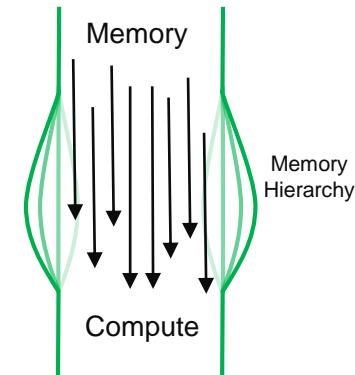
1X	1X
2X	10X
80X	100X



Fixed



Adaptable



- > Rigid memory hierarchy & data duplication
- > High “data locality” required for workload efficiency

- > Adaptable memory hierarchy & datapath
- > ~5X more on-chip memory / less off-chip required

Fixed Memory Hierarchy & Shared Interconnect:
Robs Bandwidth / Capacity & Stalls Compute

Match Memory Hierarchy & Bandwidth
to Compute Requirements

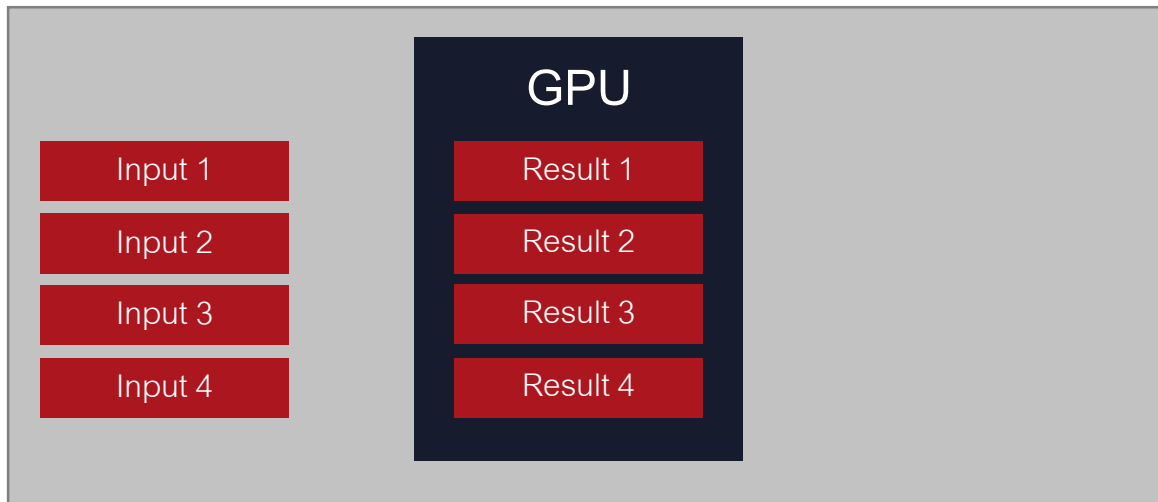
What About Batching?

Fundamental to GPU Architecture

(Software Defined Data Flow)

Batching: Loading up lots of similar Data Sets

- Keep compute cores busy
- Hide some memory latency
- Create better SIMT efficiency



High Throughput OR Low Latency

Not Required for FPGA / ACAP

(Hardware Defined Data Flow)

Independent of Data Set count

- Custom HW kernels
- Custom Memory Hierarchy
- HW pipeline data flow

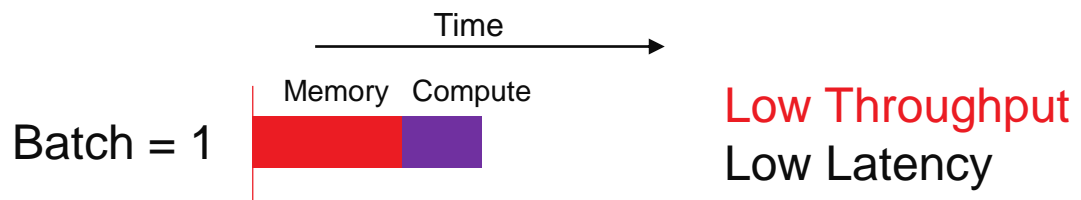
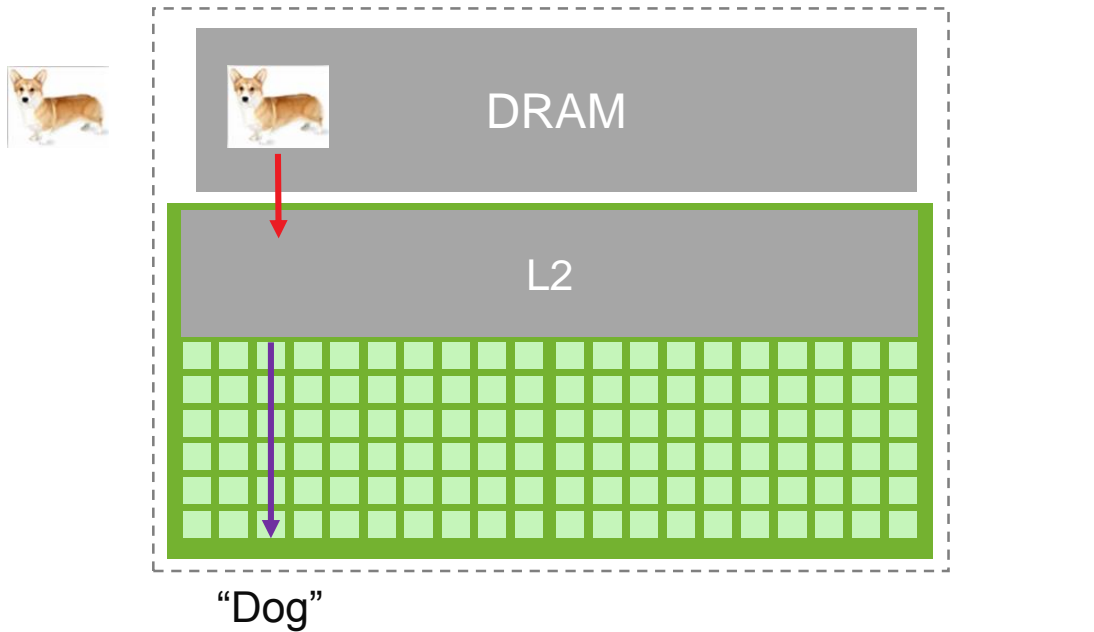


High Throughput AND Low Latency

Batching doesn't even really apply in an FPGA

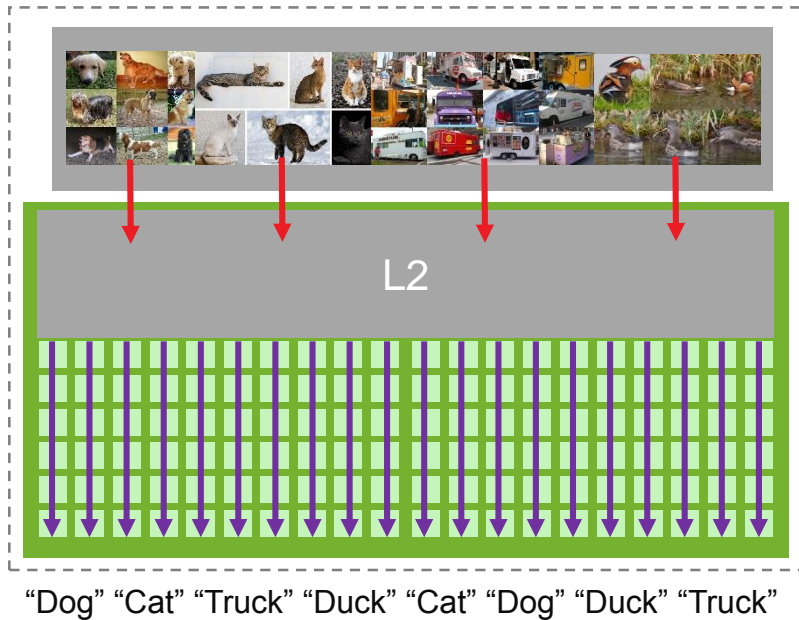
A Batching Example: Image Classification

GPU

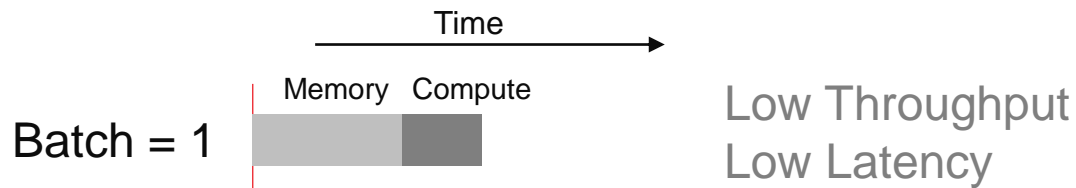
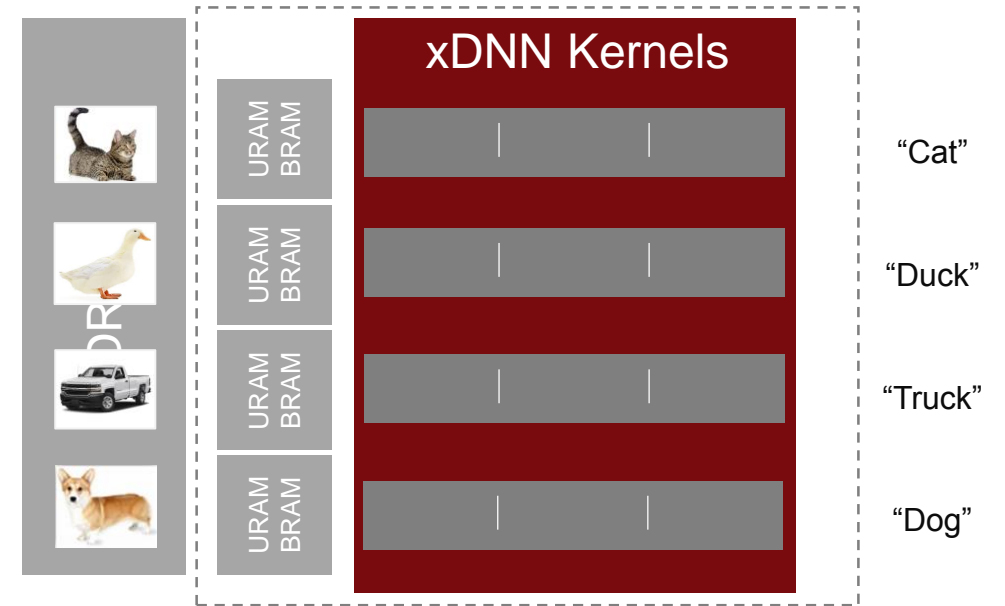


A Batching Example: Image Classification

GPU



FPGA



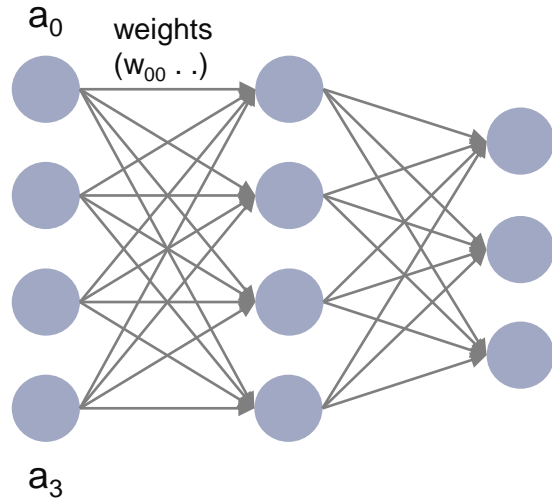
High Throughput

AND

Low Latency

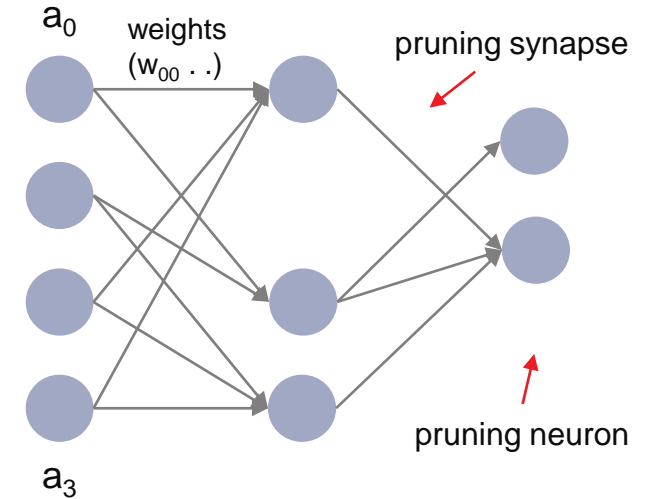
The Benefits of Pruning & Compression

Before Pruning



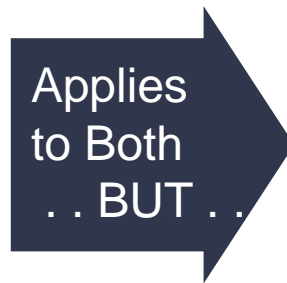
$$\begin{pmatrix} W_{00} & - & W_{02} & - \\ - & - & W_{12} & W_{13} \\ W_{20} & - & - & W_{23} \\ W_{30} & - & - & W_{33} \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} ? \\ ? \\ ? \\ ? \end{pmatrix}$$

After Pruning



Pruning Benefits:

- Smaller, “lighter” networks
- Less mem capacity & b/w req'd
- Reduced compute requirement
- Higher performance
- Lower power



GPU

- Issues w/ sparse & small matrices
- Compute efficiency degrades
- Still more off-chip mem req'd

FPGA

- Better w/ sparse matrices
- Single-chip solutions possible
- Device scales w/ compute/resources

✓ Up to 30+% Better Compression

Only Adaptable Hardware Addresses Inference Challenges

Custom data flow
(Address new architectures)



Custom memory hierarchy
(Address power/performance challenges)



Custom precision
(Address power/cost)



Domain Specific
Architectures (DSAs)
on Adaptable Platforms



Do TOPs/FLOPs Matter?

Device Figures of Merit

TOPS & TFLOPS



33 TOPS

8 TFLOPS

✓✓ Availability
In Data Sheets

✗ Usability

Mainly Useful for Relative
Comparison within a Product Line

Putting Metrics & Benchmarks in Focus

Usable TOPS limited by:

- Memory bottlenecks
- Code / data structure
- Stalls & branches
- Freq. throttling

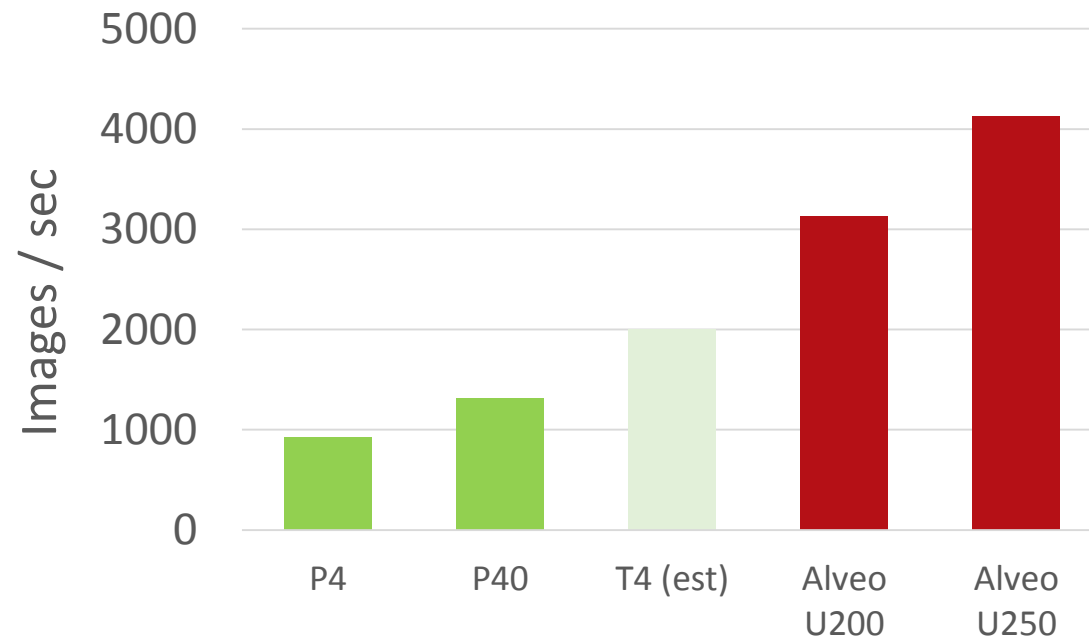
AND Application Reqs:

- Request ("Batch") Size
- Latency Spec
- Power Envelope
- Accuracy Requirements



ML Benchmark

GoogleNetv1 Batch=1 Throughput

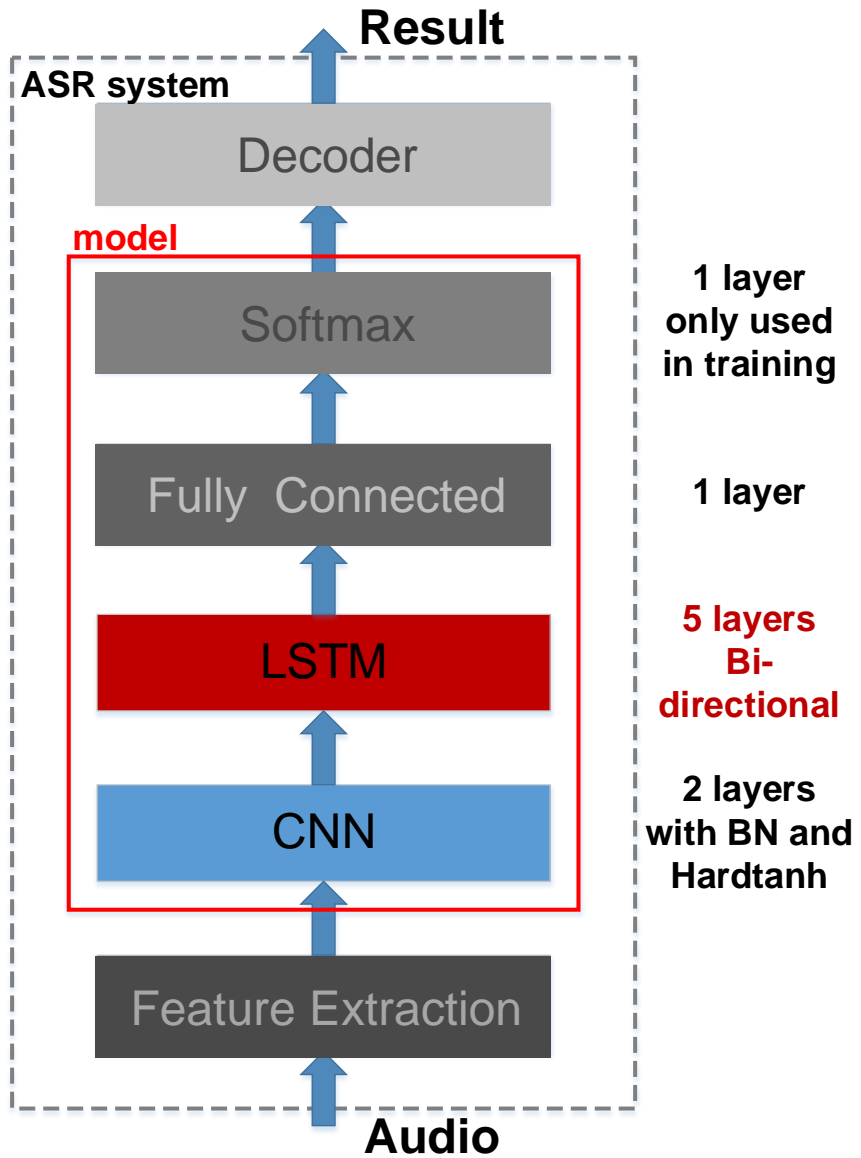


✓ Adaptable Hardware Delivers

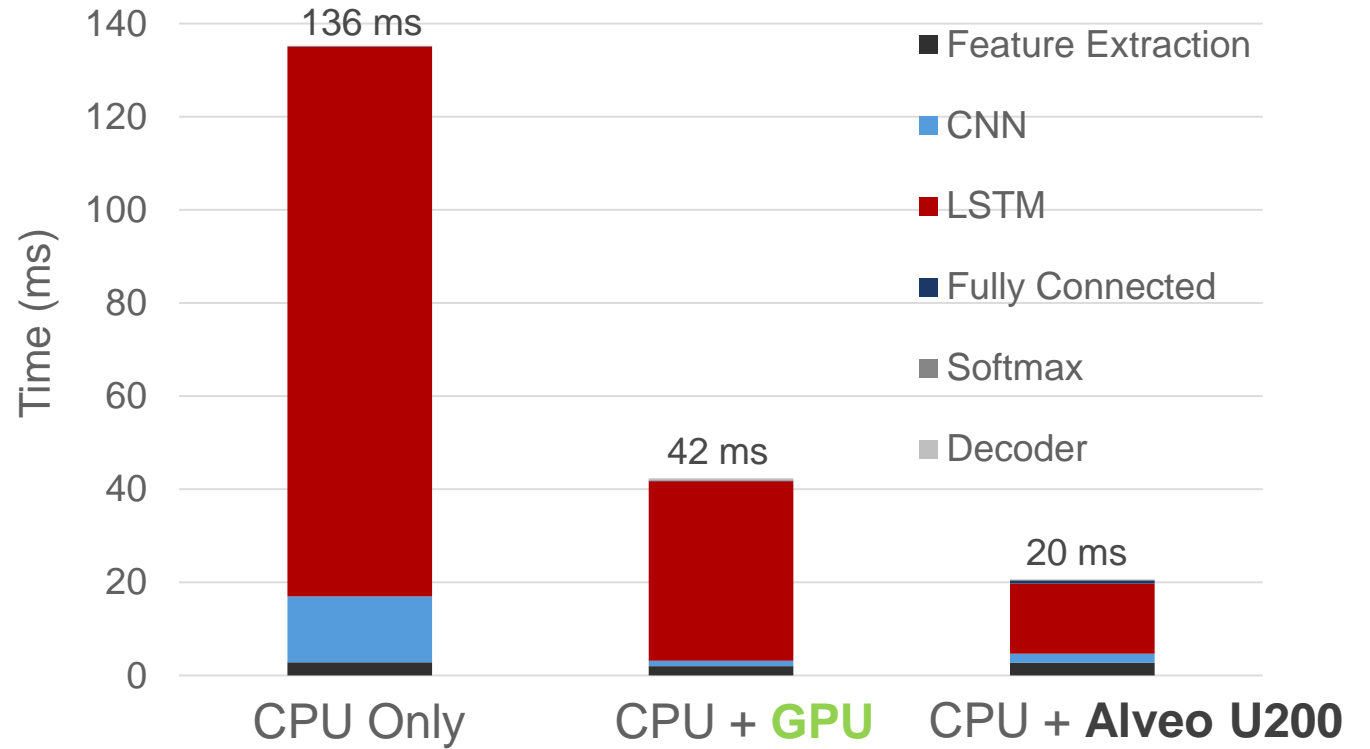
- Custom compute / memory
- Higher device utilization
- HW performance / power

Focus on Application Level Performance Where Xilinx Solutions Shine

Bi-directional LSTM Performance: Speech to Text



End-2-End Time: AWS Instance Comparison

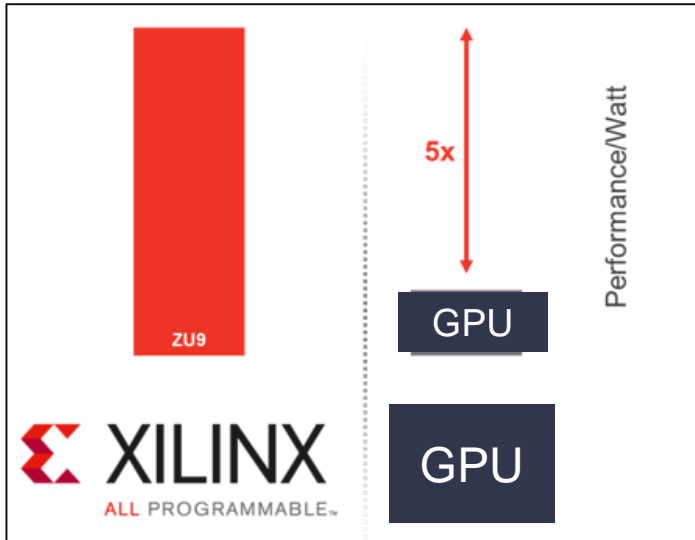


VU9P Delivers Fastest E2E Time

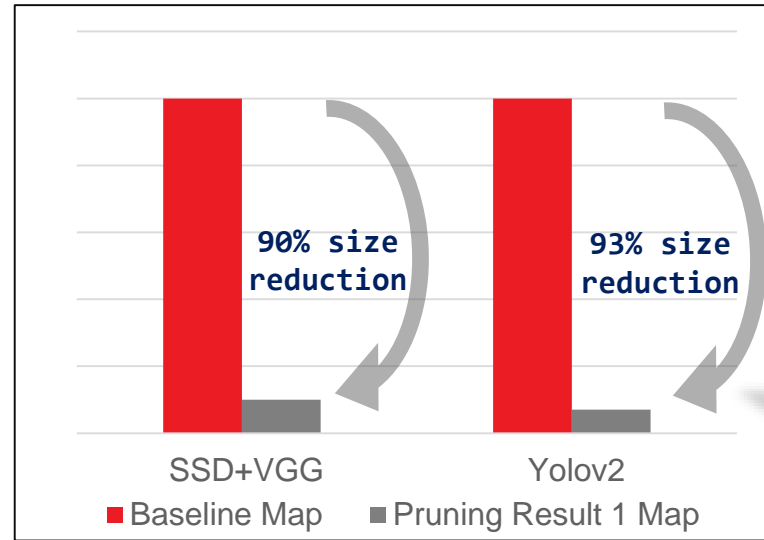
8X vs. CPU / **2X** vs. GPU

Xilinx Machine Learning Customer Successes

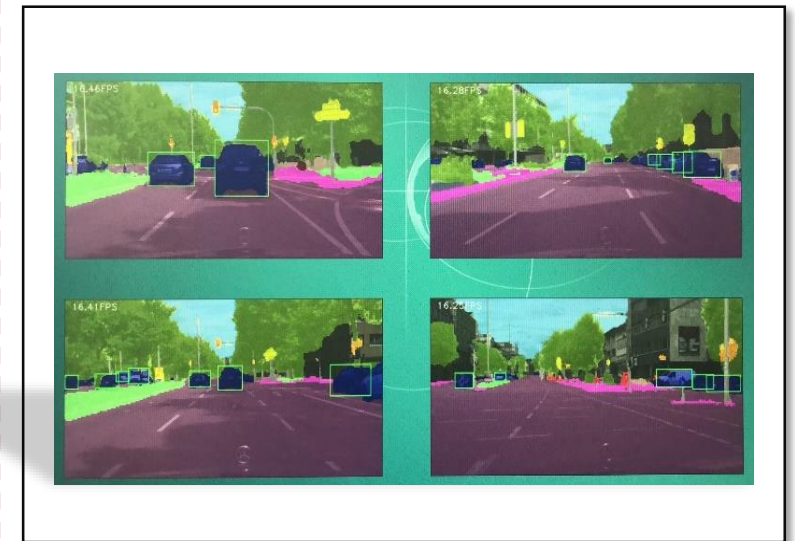
Surveillance System



Automotive OEM



Automotive OEM



- USB Camera +CV +ML +Display
- **5x better Perf/Watt than GPU / SSD (no pruning!)**

- ML benchmarks (pruning)
- **93% Reduction of resources within 1% of initial precision**

- Camera +CV +12 SSD +Display
- **12 channel object detection in 1 ZU9 (with pruning)**

Video Success Story

Conclusion



FPGAs Address the Machine Learning Challenges . . .



The rate of AI innovation



Performance at low latency



Low power consumption



Whole app acceleration

. . . and today's activities will show you exactly how.

Thank you.



Adaptable **Advantage**

