DnnWeaver: From High-Level Deep Network Models to FPGA Acceleration

Hardik Sharma Jongse Park Divya Mahajan Emmanuel Amaro

Joon Kyung Kim Chenkai Shao Asit Mishra[†] Hadi Esmaeilzadeh

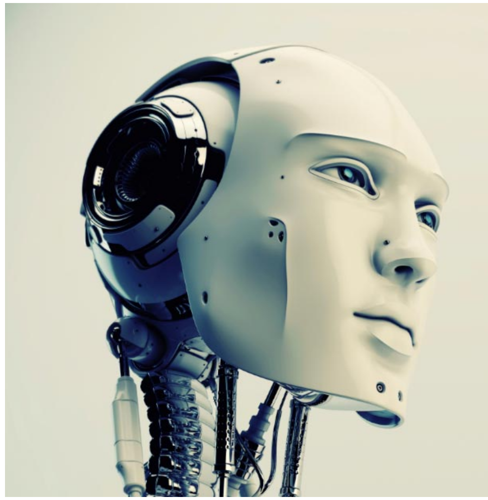
Alternative Computing Technologies (ACT) Lab
Georgia Institute of Technology

[†]Intel Corporation

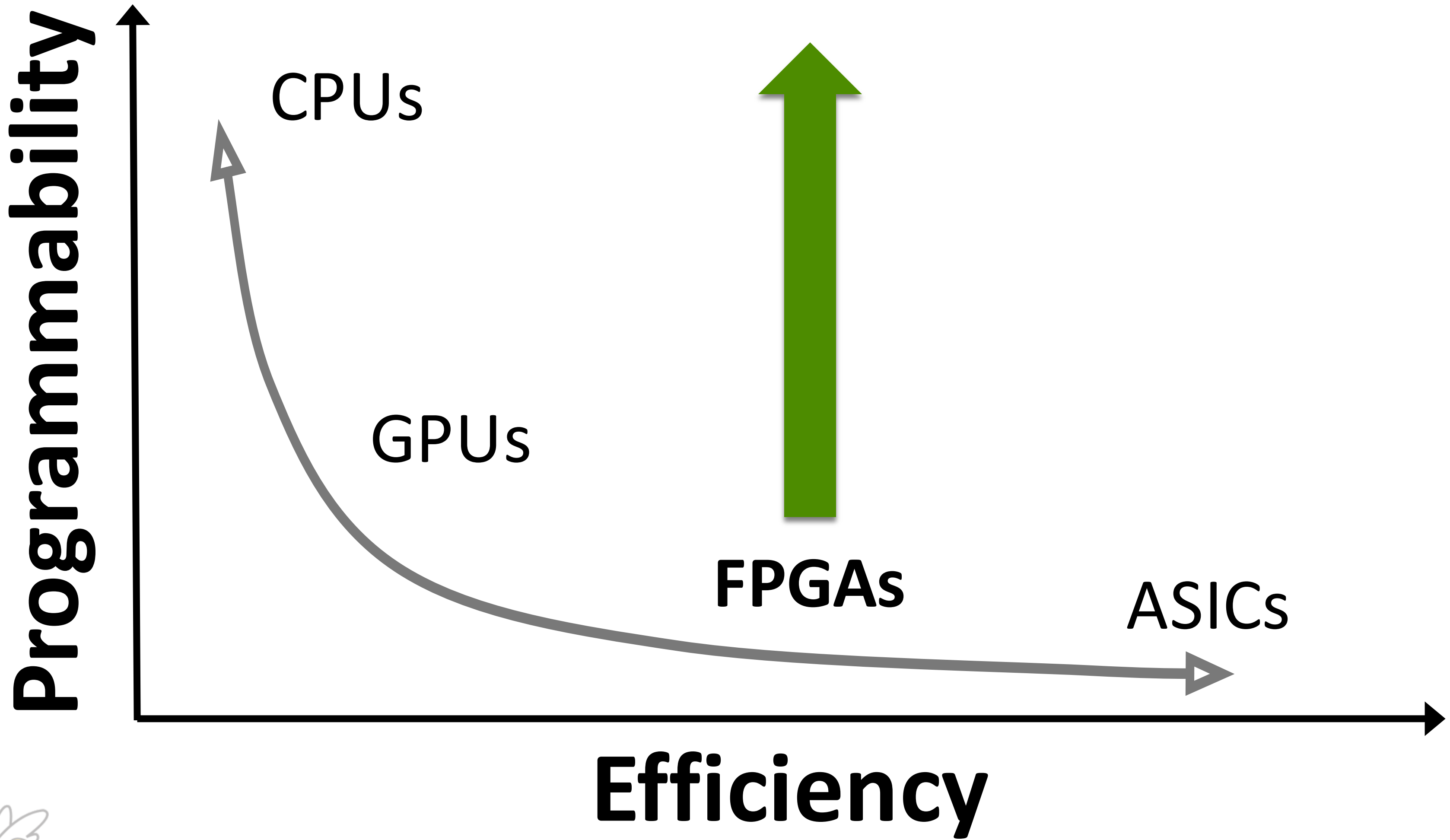
MICRO'16

Video Categorization

Deep Neural Networks: A Move Towards Artificial Intelligence



Programmability is a First-Order Concern



Different applications require different DNNs!

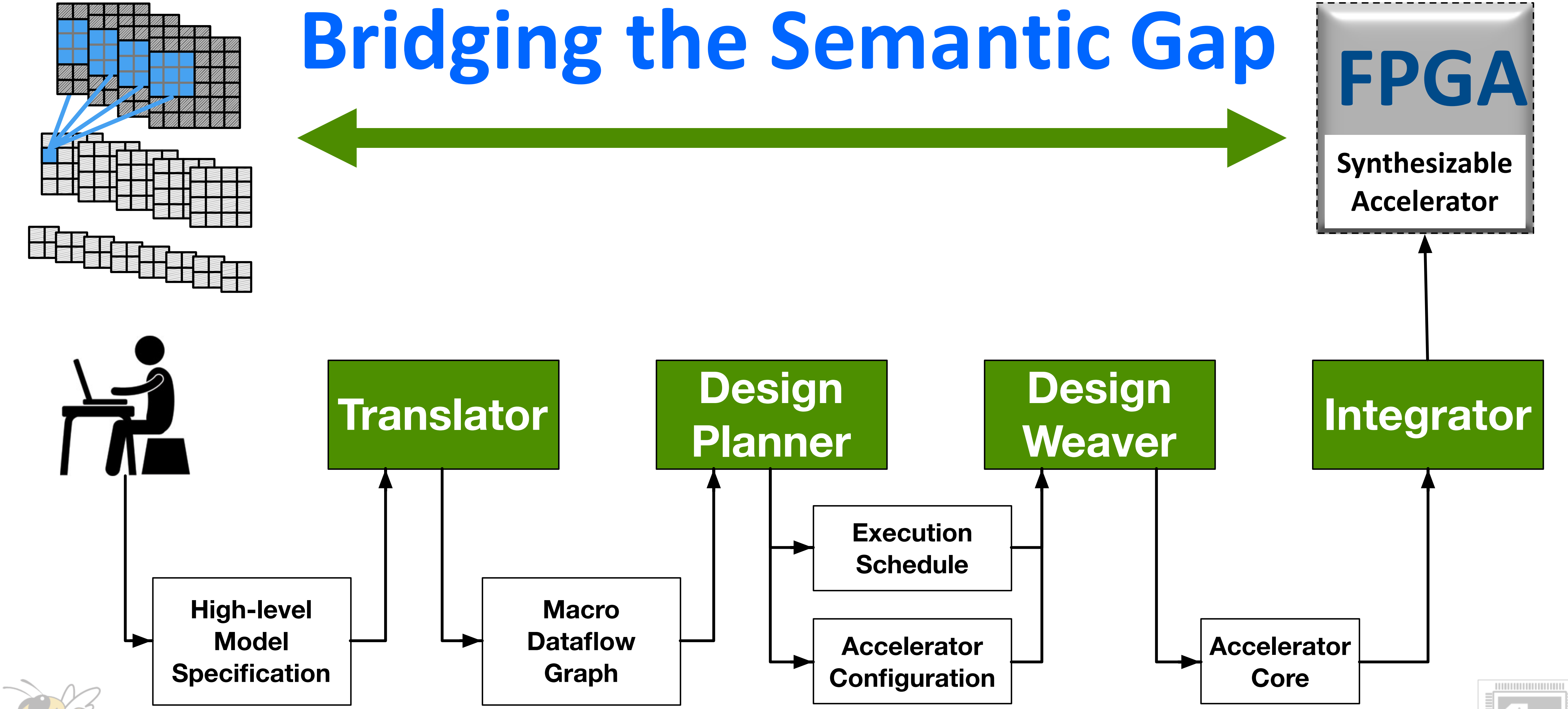
FPGAs are Hard to Program!

Over 10,000 lines of code for DNN hardware templates in 14 months!

DNNs are Big, on-chip FPGA storage is small!

AlexNet is 119 MBytes whereas Arria 10 has 5 MB!

Bridging the Semantic Gap



FPGA

Synthesizable Accelerator

Translator

Design Planner

Design Weaver

Integrator

High-level Model Specification

Macro Dataflow Graph

Execution Schedule

Accelerator Configuration

Accelerator Core

1. Translator

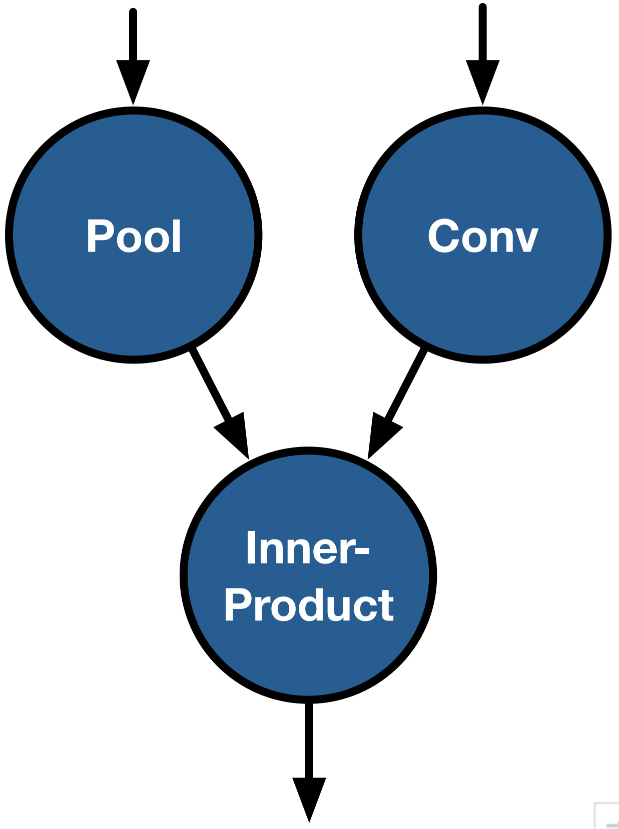
High-Level DNN model – Caffe*

```
layer{
name: Pool,
type: POOLING,
params{…}}

    layer{
    name: Conv,
    type: CONVOLUTION,
    params{…}}

        layer{
        name: Inner-Product,
        type: INNER_PRODUCT,
        params{…}}
```

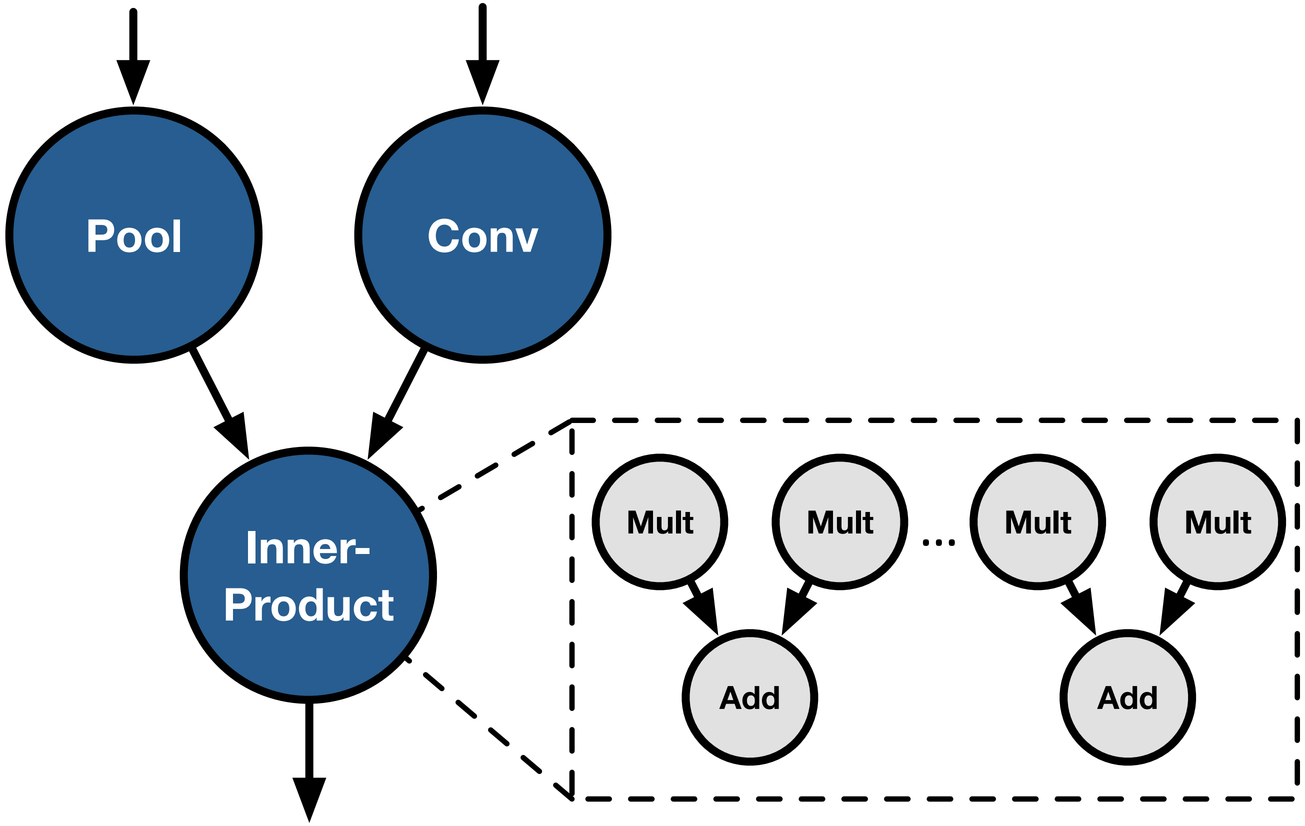
DnnWeaver ISA



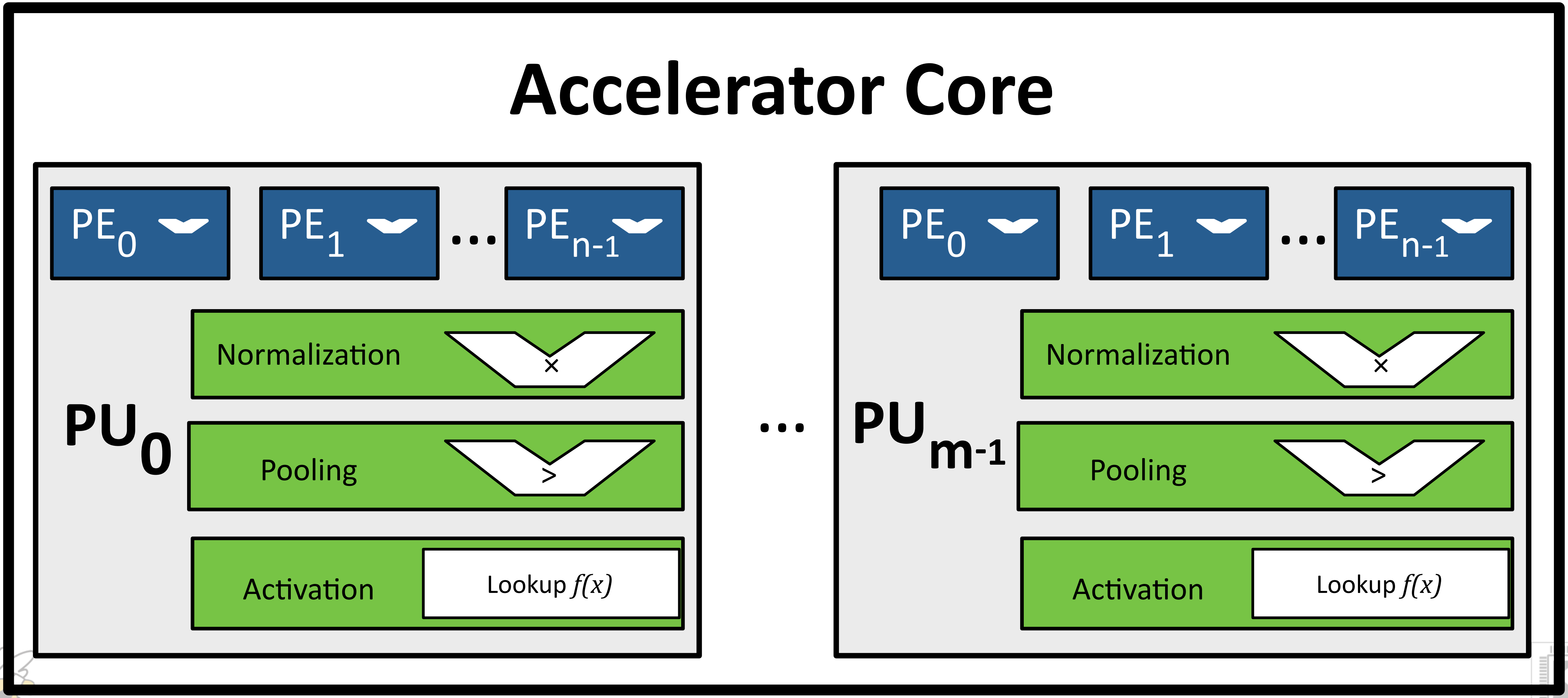
*Caffe: Convolutional Architecture for Fast Feature Embedding. arXiv:1408.5093

ISA: Abstracting DNNs

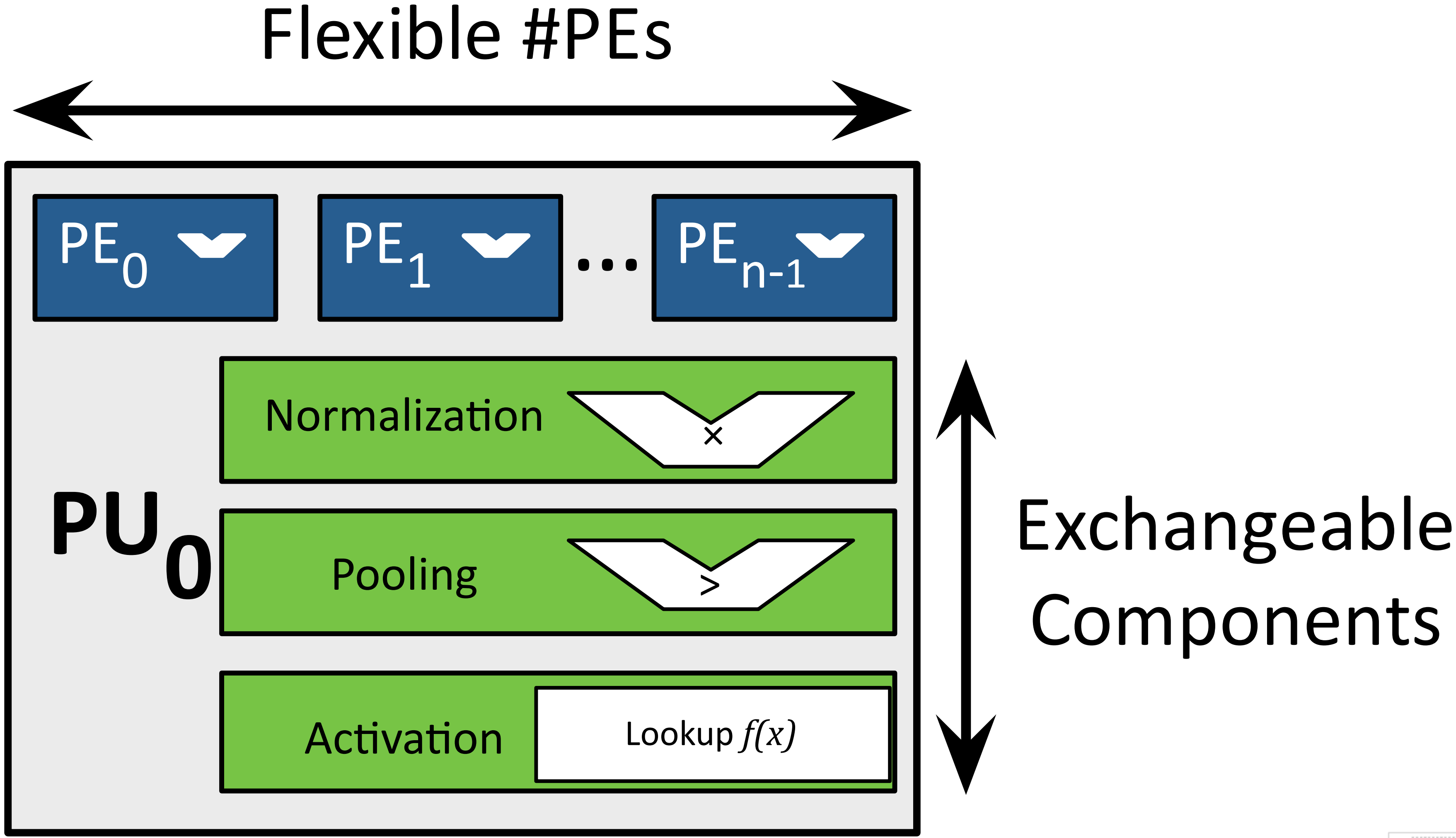
Abstract DNNs as a **macro-dataflow graph**



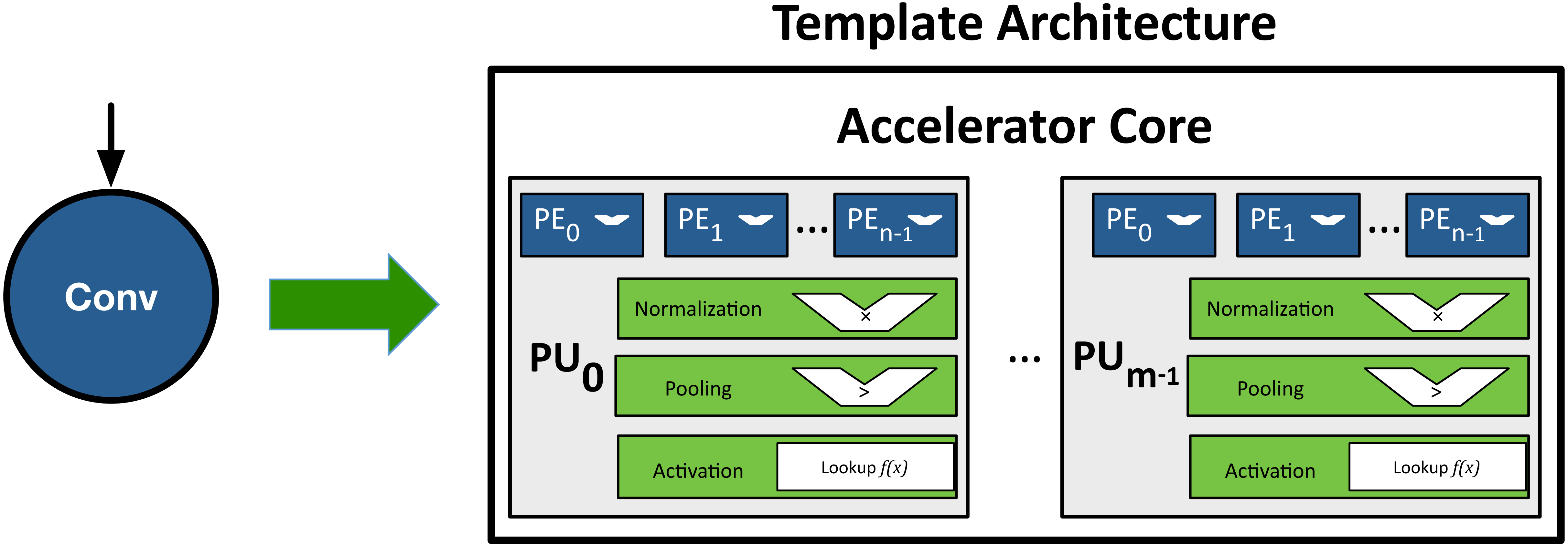
Template Architecture



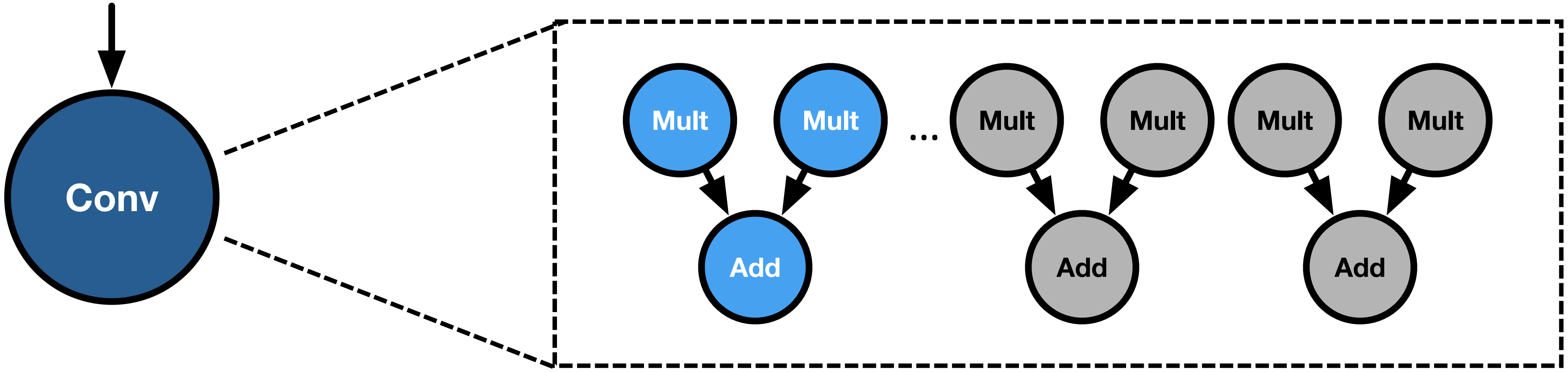
Processing Unit

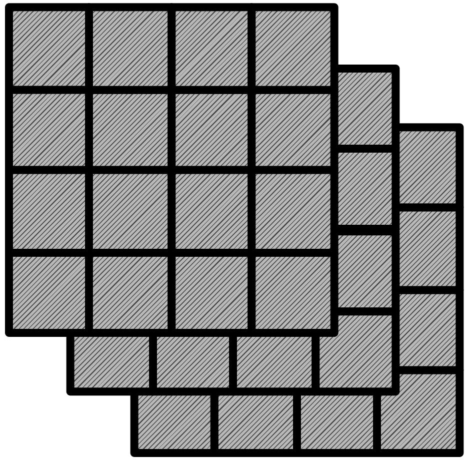


Executing the Macro Dataflow Graph

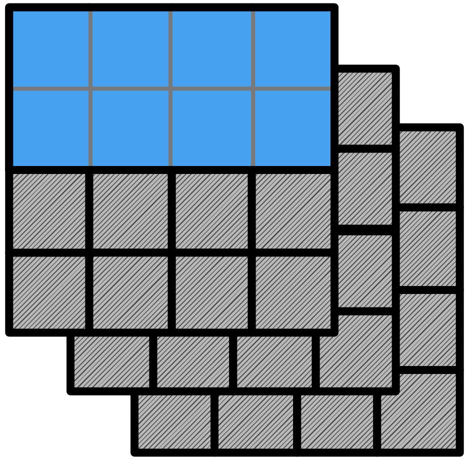


Slicing the Dataflow Graph





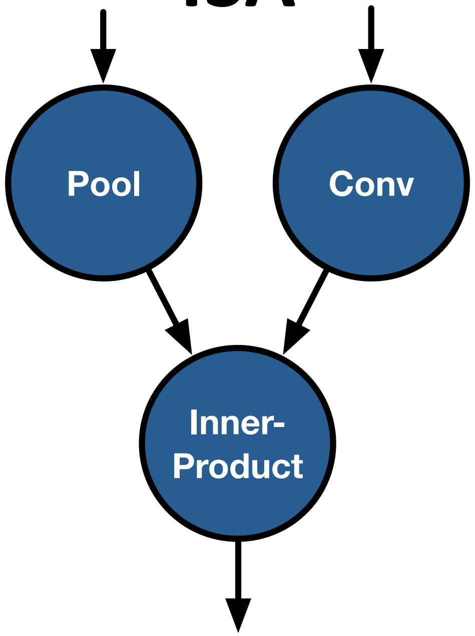
Convolution Node Output



Sliced Output

2. Design Planner

Macro-Dataflow ISA



FPGA Specification

# of DSPs
# of BRAMs
BRAM capacity
# LUTs

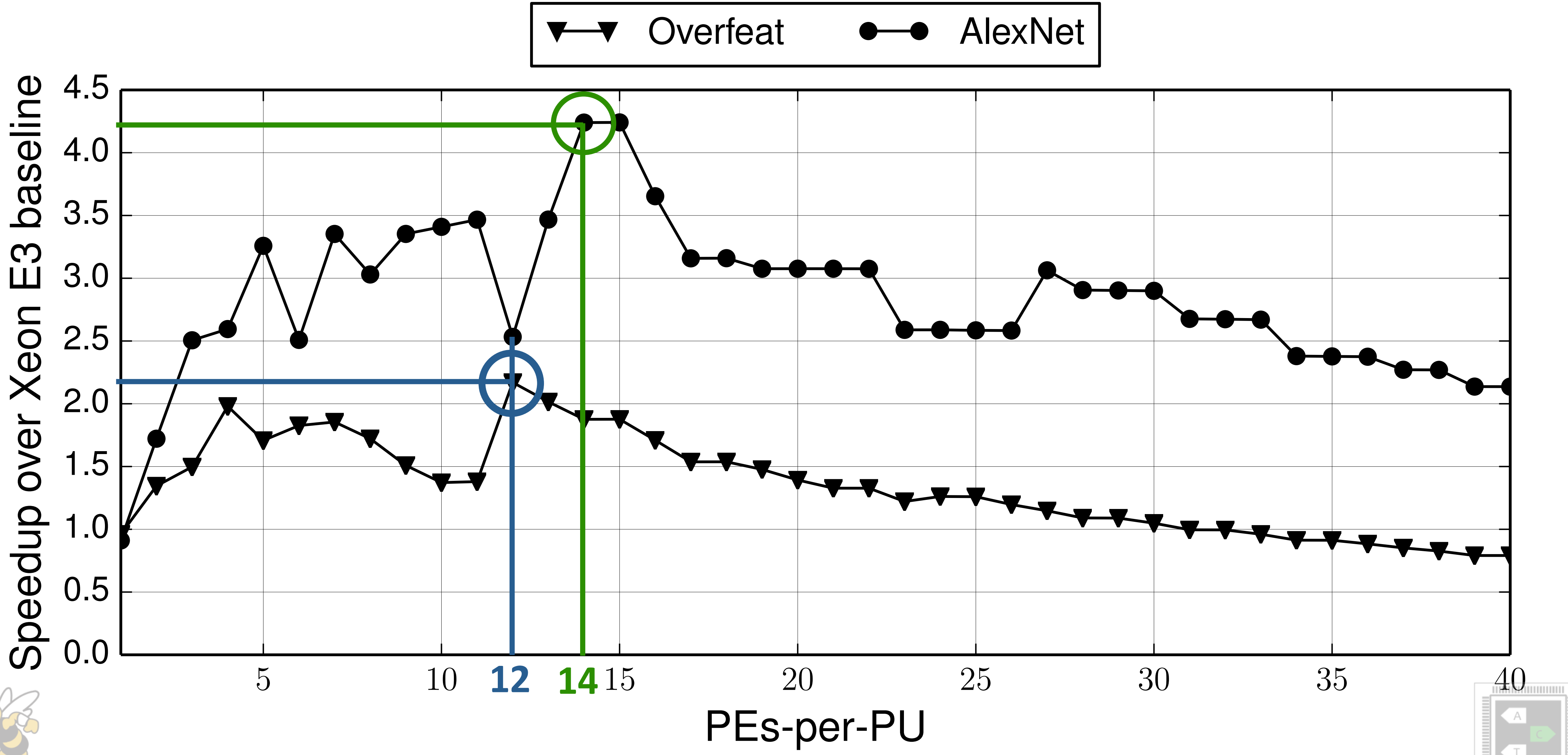
Co-optimize Hardware and Execution Schedule

Using FPGA’s Programmability

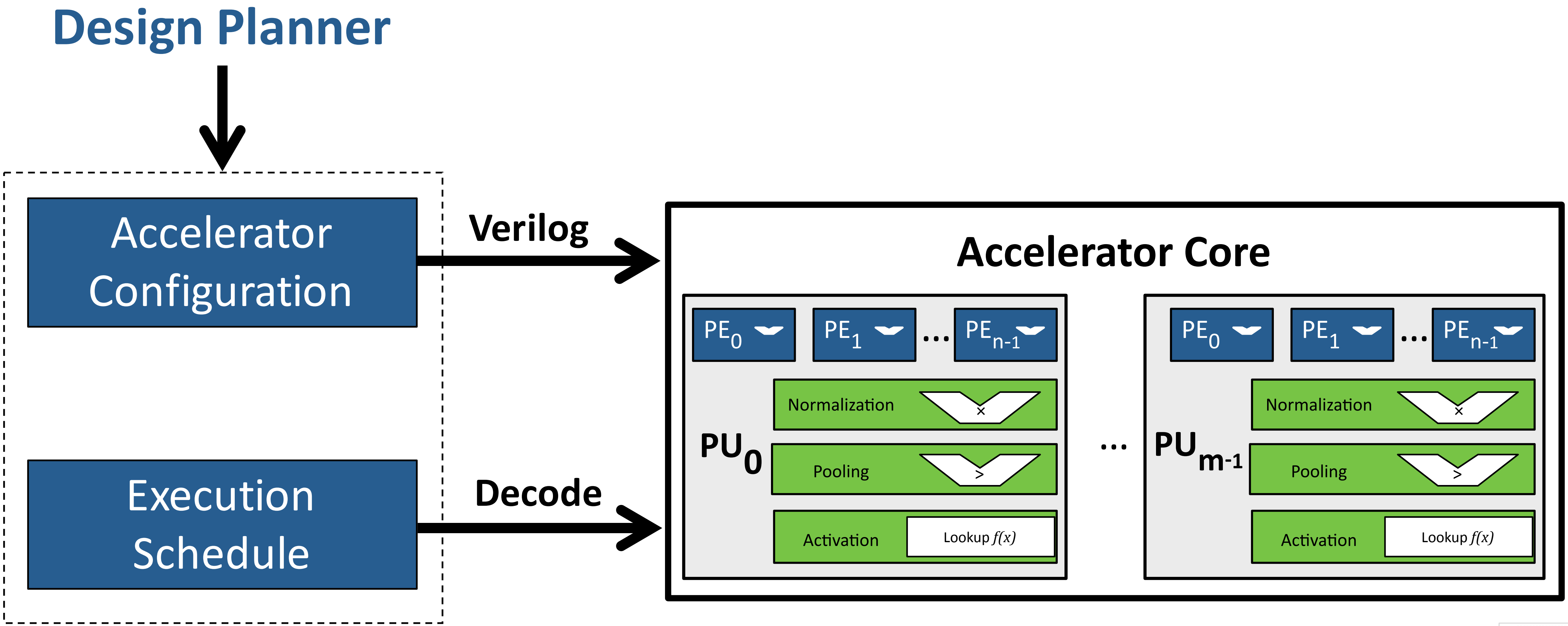
Significant variations between different DNN models: # layers, Model Size, Operations, etc.

Specialize for each DNN model!

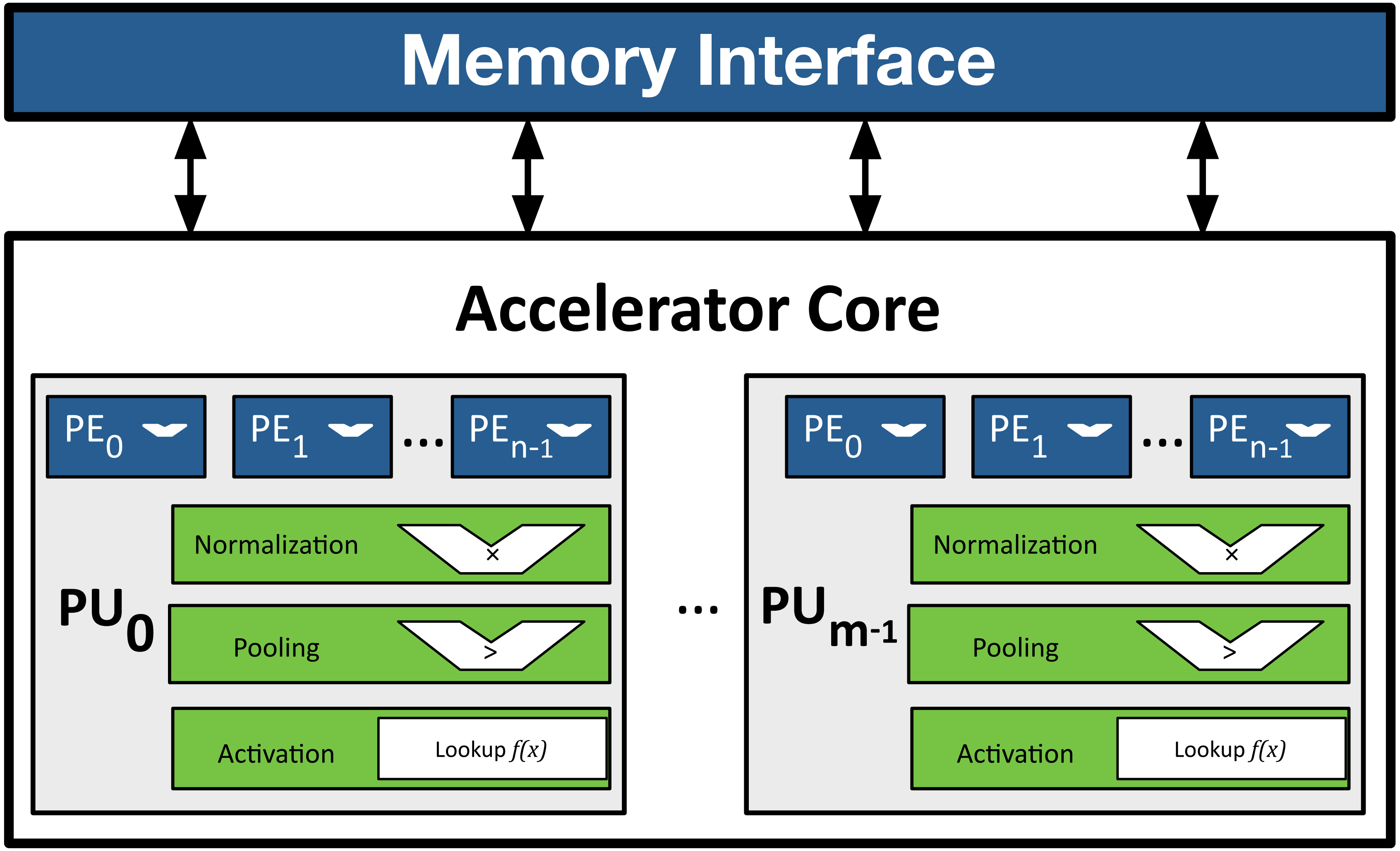
Co-optimization



3. Design Weaver



4. Integrator



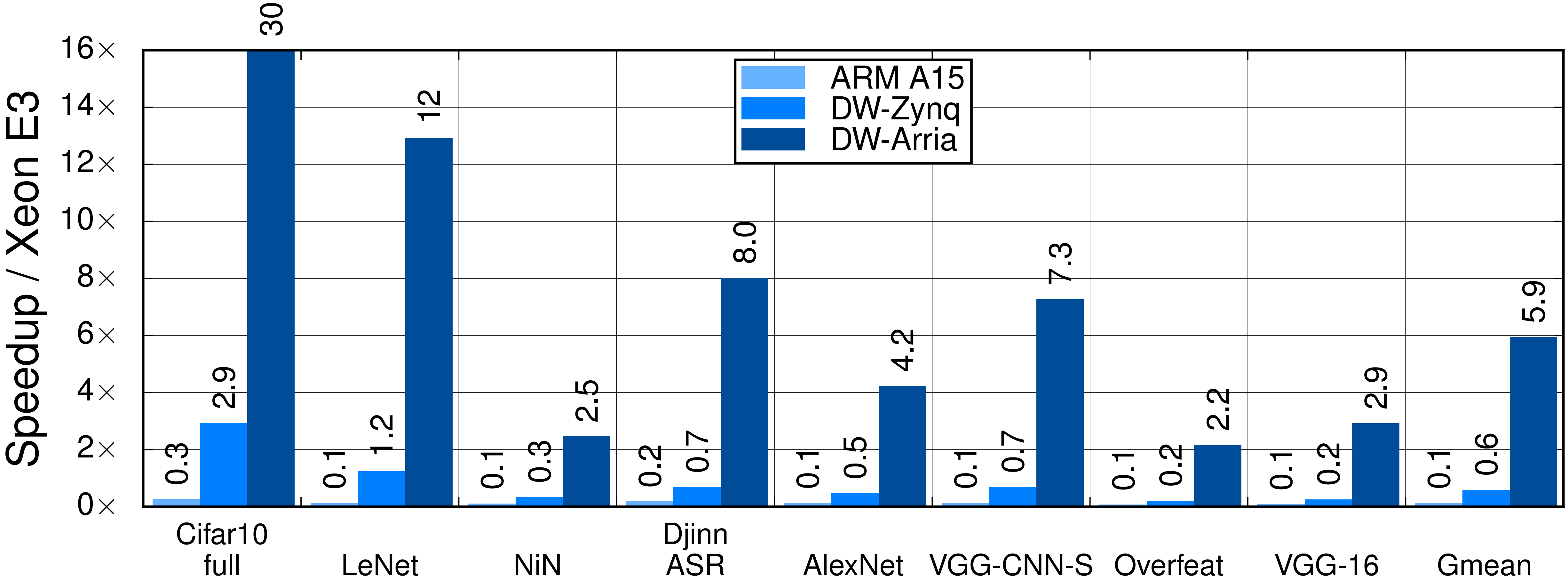
Benchmark DNNs

Name	# Layers	Model Size(MB)	# Operations (Mops)	Lines of Code
VGG-16	36	324.0 MB	16362 MOps	347
OverFeat	16	278.0 MB	2798 MOps	196
VGG-CNN-S	19	196.0 MB	2666 MOps	200
AlexNet	20	119.0 MB	1147 MOps	278
Djinn	13	48.4 MB	25 MOps	105
NiN	28	14.5 MB	1106 MOps	516
LeNet	7	0.8 MB	2 MOps	128
CIFAR-10Full	12	0.2 MB	12 MOps	156

Platforms Tested

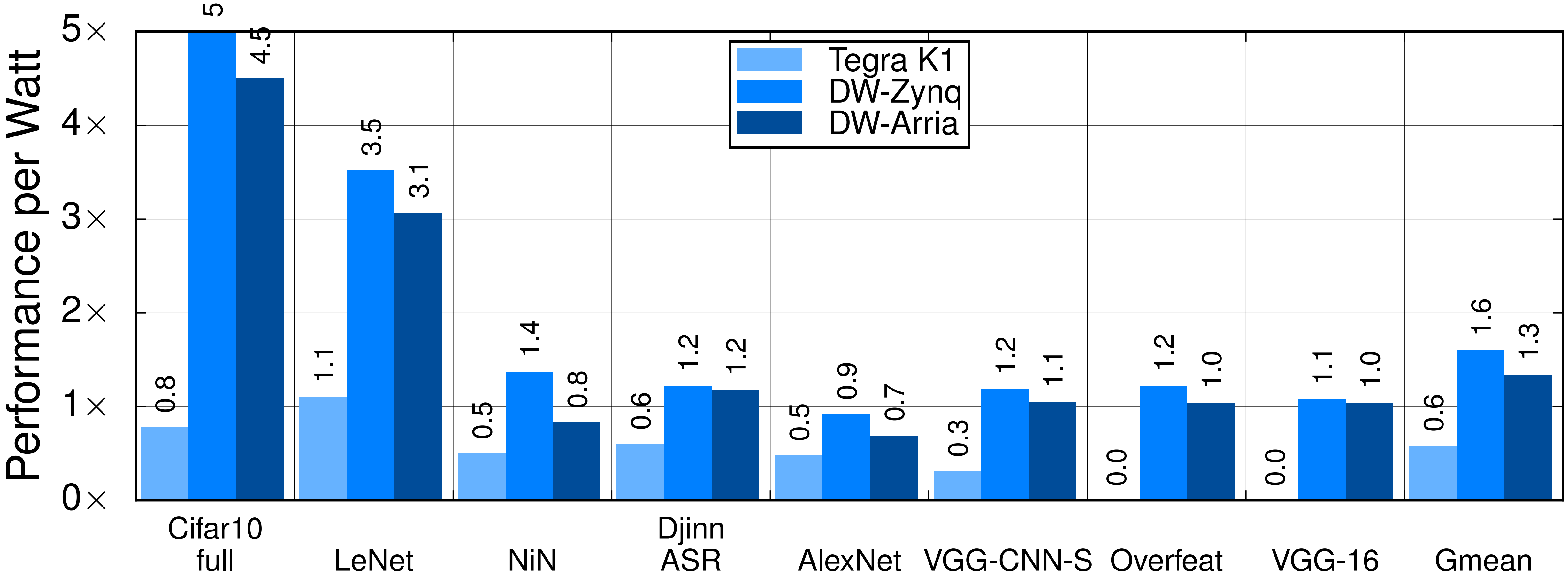
	High Performance		Low Power
FPGA	Altera Arria 10	Altera Stratix V	Xilinx ZC702
CPU	Intel Xeon E3		ARM Cortex 15
GPU	Tesla K40	GTX 650 Ti	Tegra K1

Performance vs CPUs



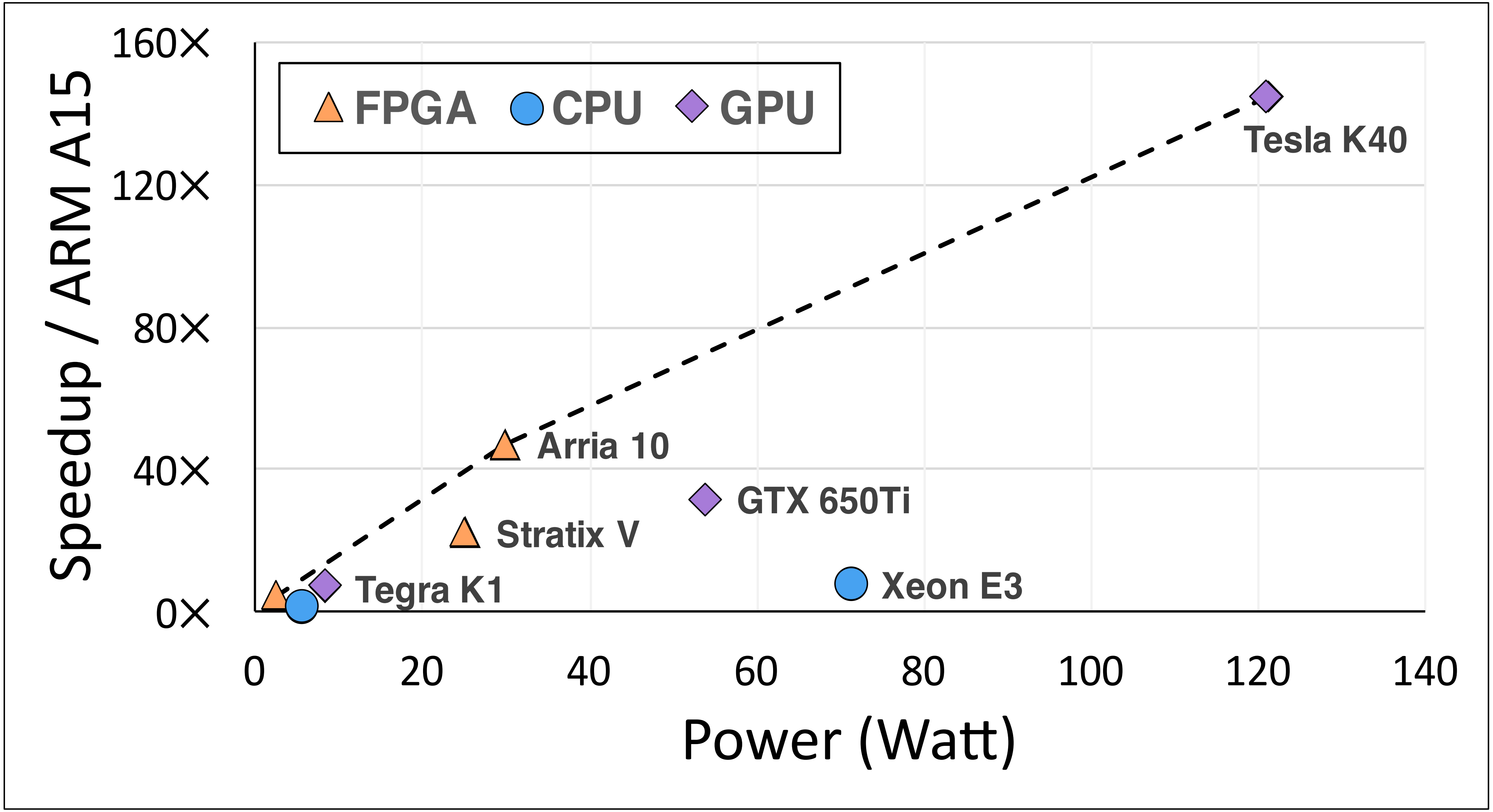
Compared to Xeon, Arria10 is **5.9x** faster, and Zynq is **0.6x** faster.

Performance-per-Watt vs GPUs



Compared to TeslaK40, Zynq is **1.6x**, and Arria10 is **1.3x** more power efficient.

Performance vs Power



Conclusion

FPGAs are a promising option for low-mid power range

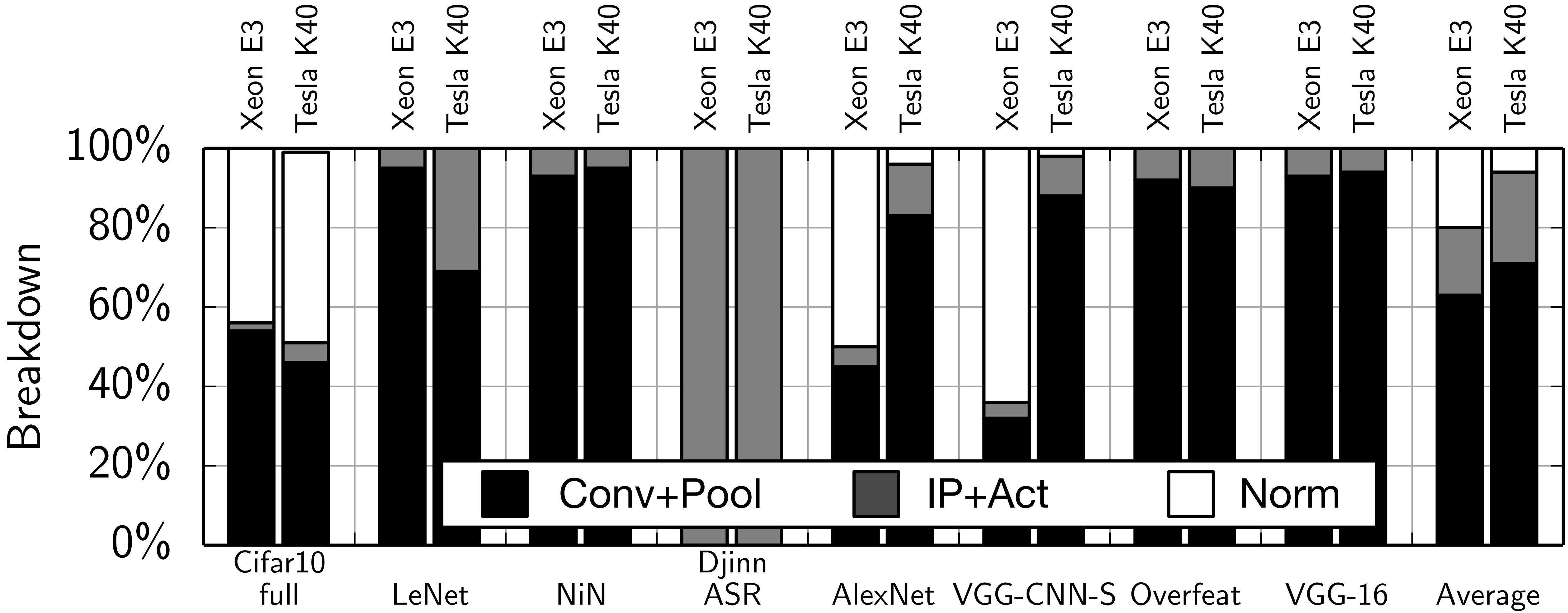
However, there is a semantic gap between the high-level DNN models and FPGA acceleration

DNNWEAVER is an initial step in making FPGAs more accessible to DNN programmers

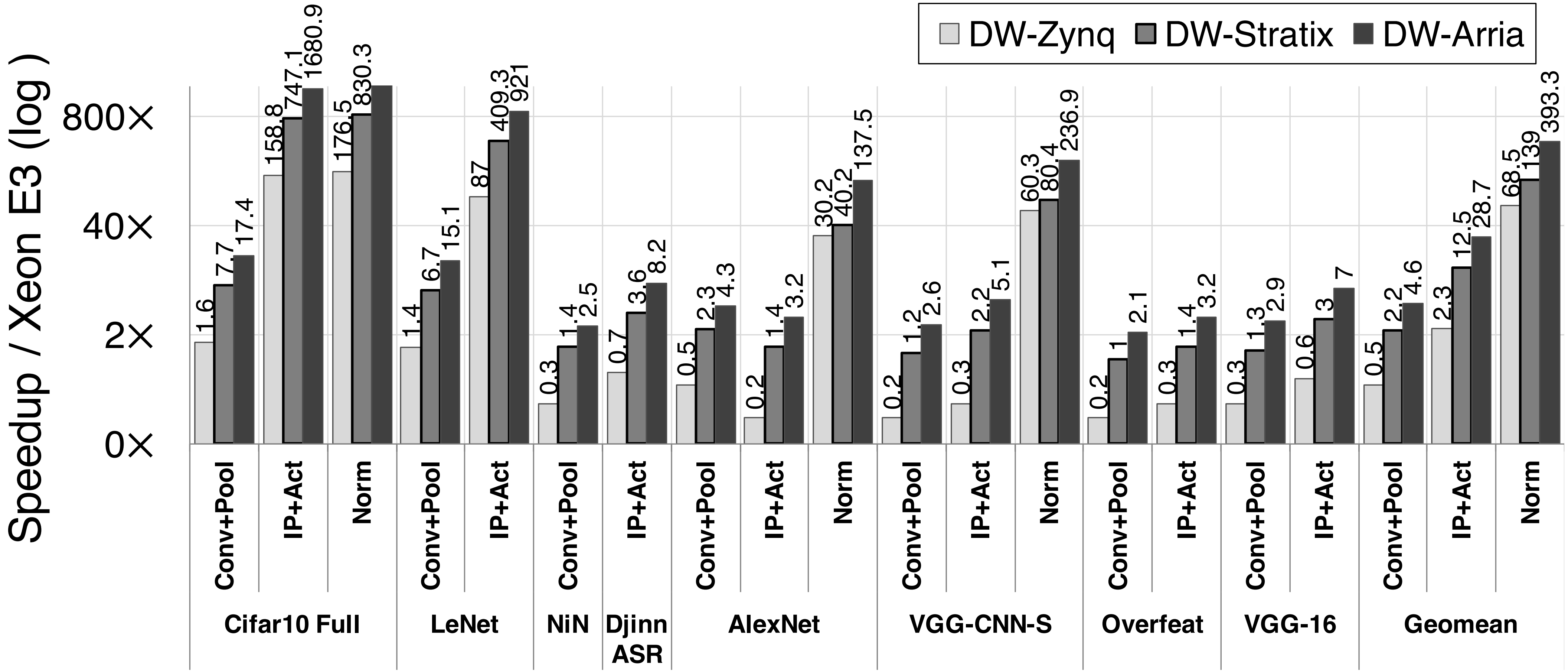
Questions?

Backup Slides

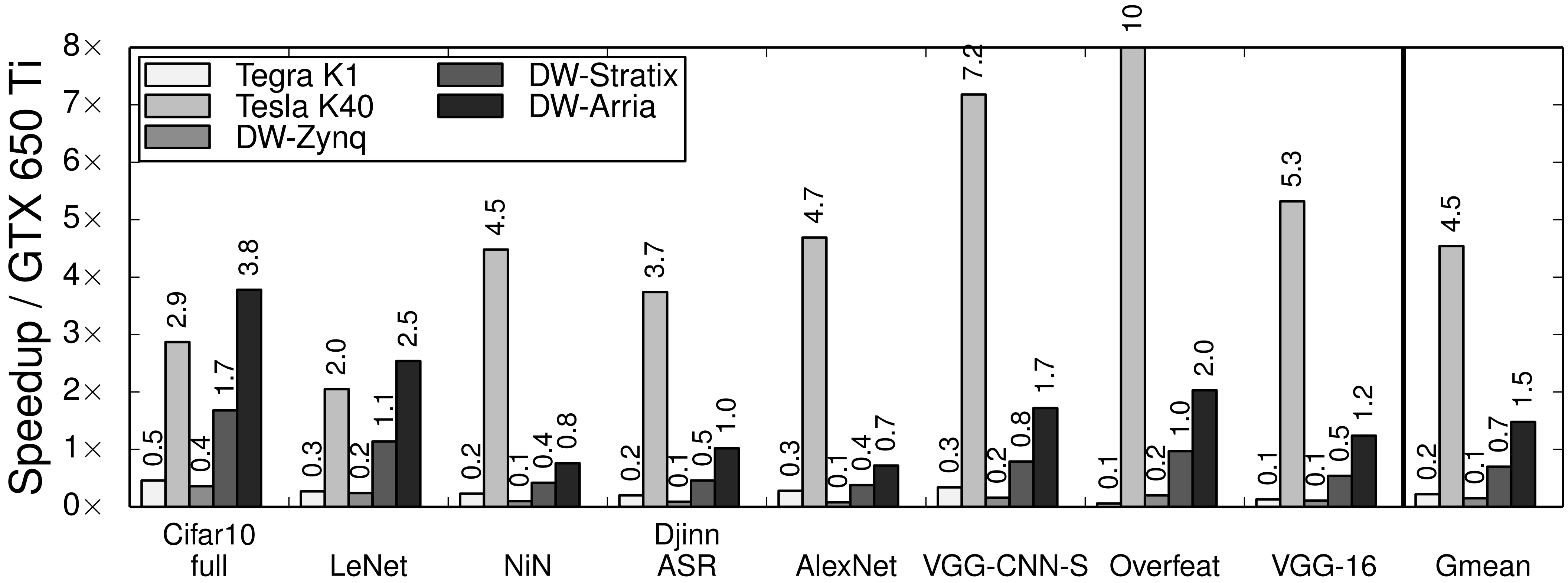
Runtime Breakdown



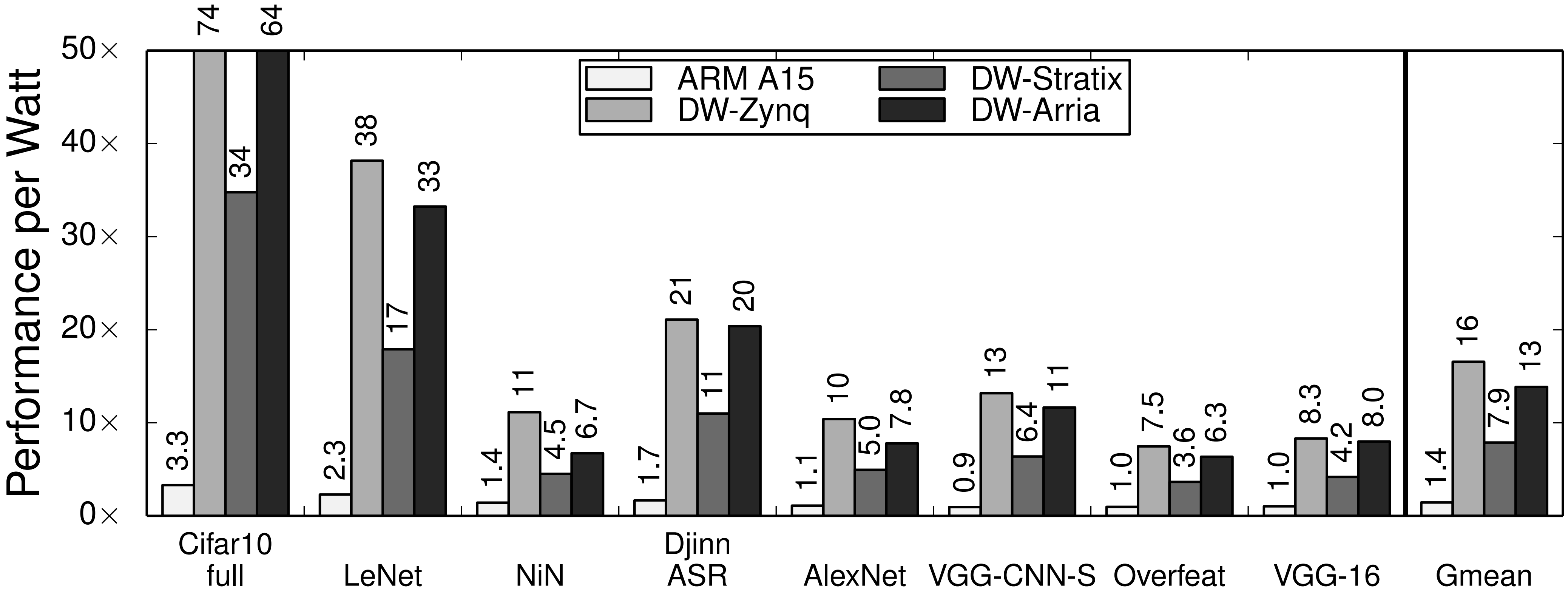
Per Layer Speedup CPU



Performance vs GPU



Performance-per-Watt vs CPU



Platforms Tested

Altera Arria 10
TDP:35W
$4495

Altera Stratix V
TDP:25W
$6999

Xilinx Zynq ZC702
TDP: 2W
$129

Intel Xeon E3-1276 V3
TDP: 84W
$339

ARM Cortex 15
TDP: 5W
$191

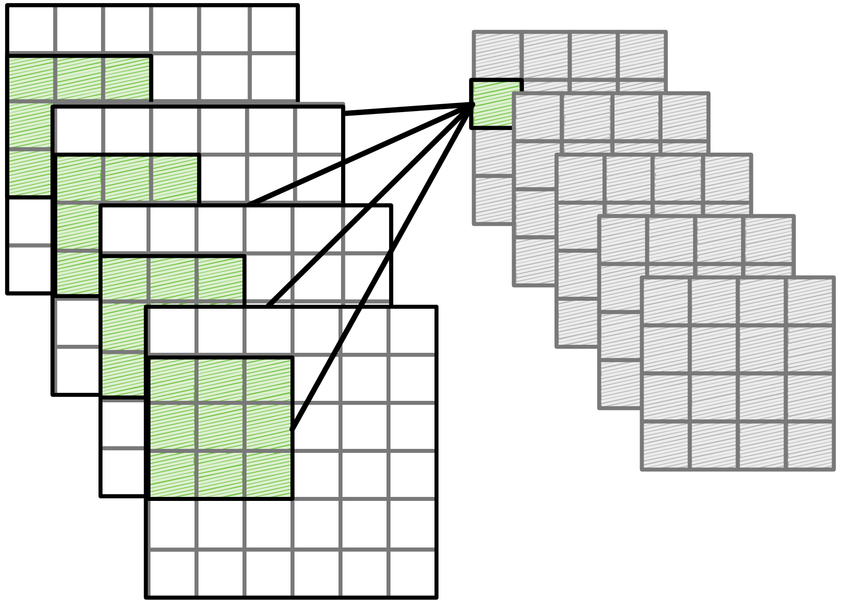
Tegra K1 GPU
TDP: 10 W
$191

GTX 650 Ti
TDP: 110
$150

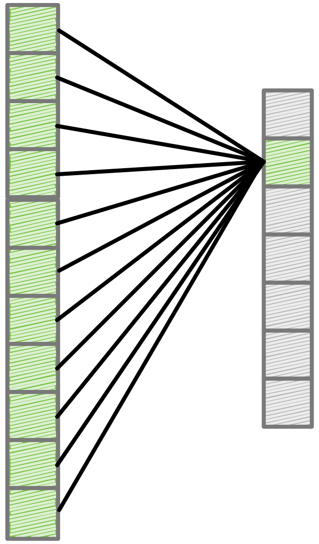
Tesla K40
TDP: 235 W
$5499

Deep Neural Networks (DNNs)

Convolution

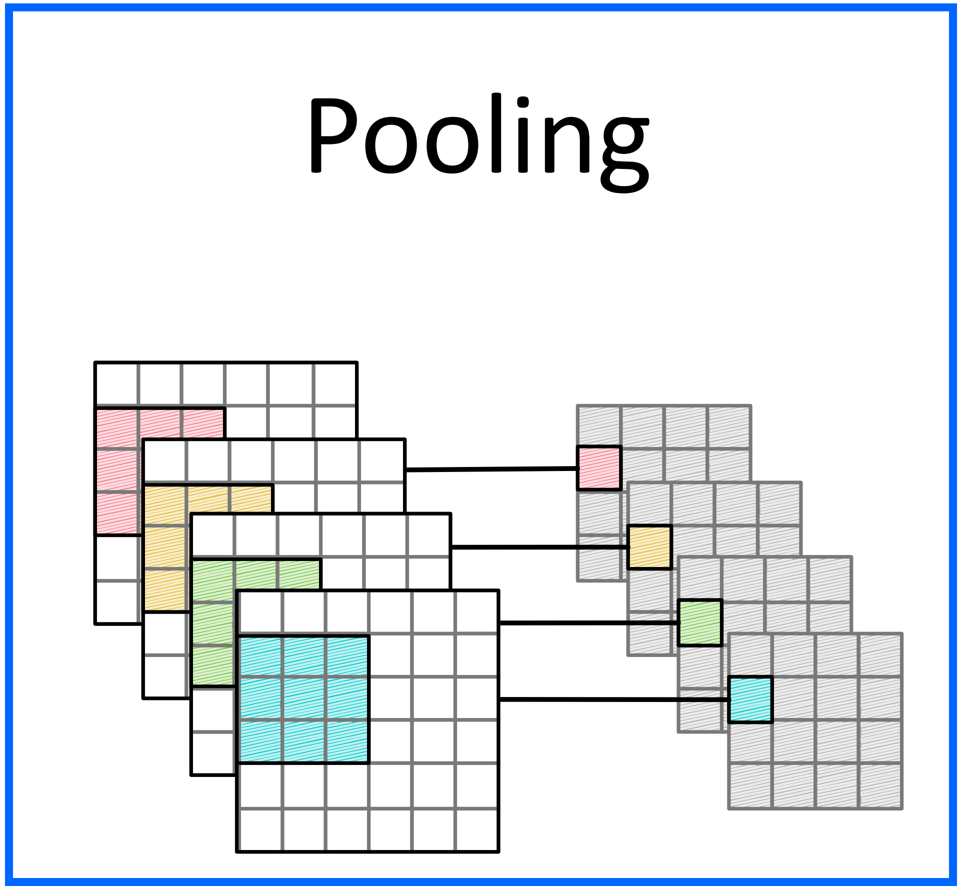


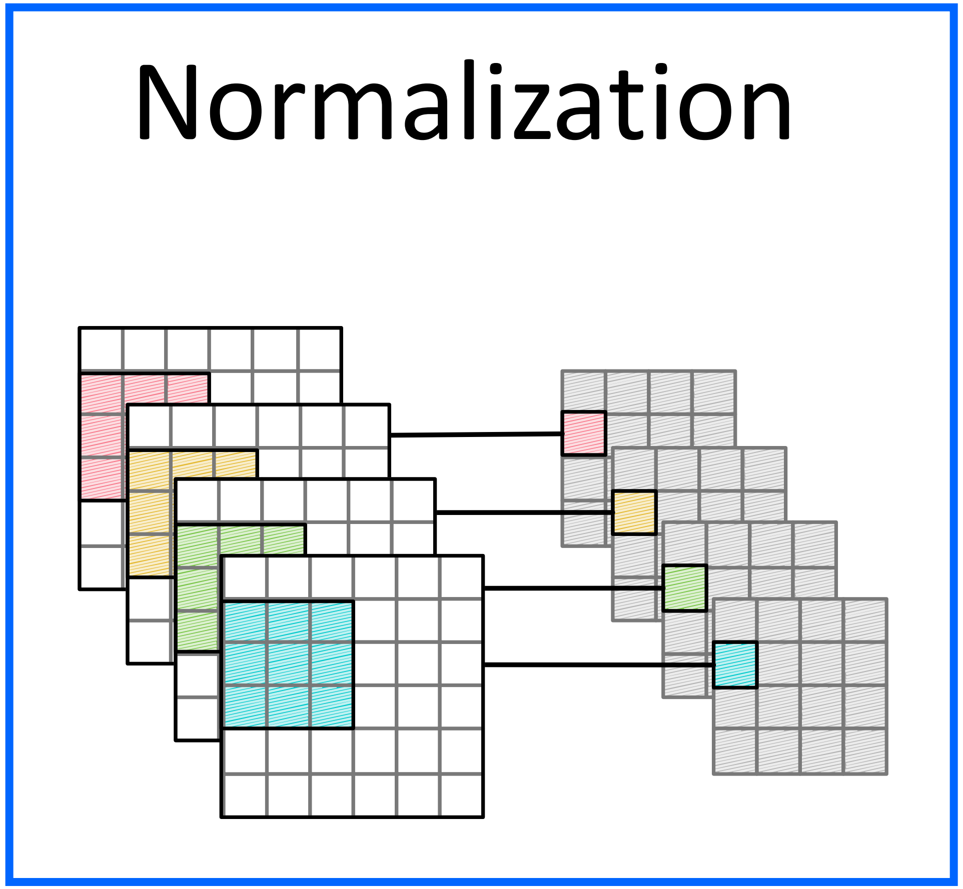
Inner-Product

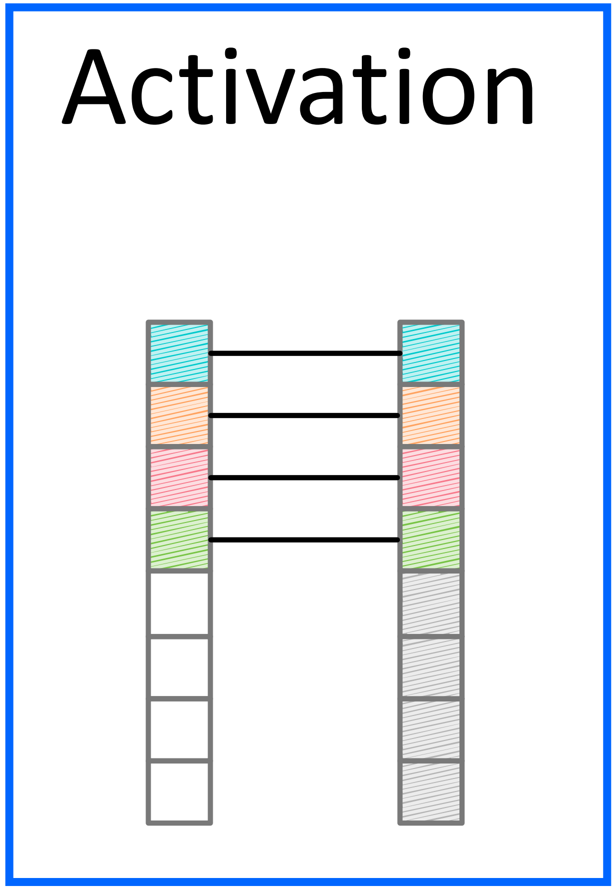


Convolution and Inner-Product are the Learnable Layers

Deep Neural Networks (DNNs)


Pooling

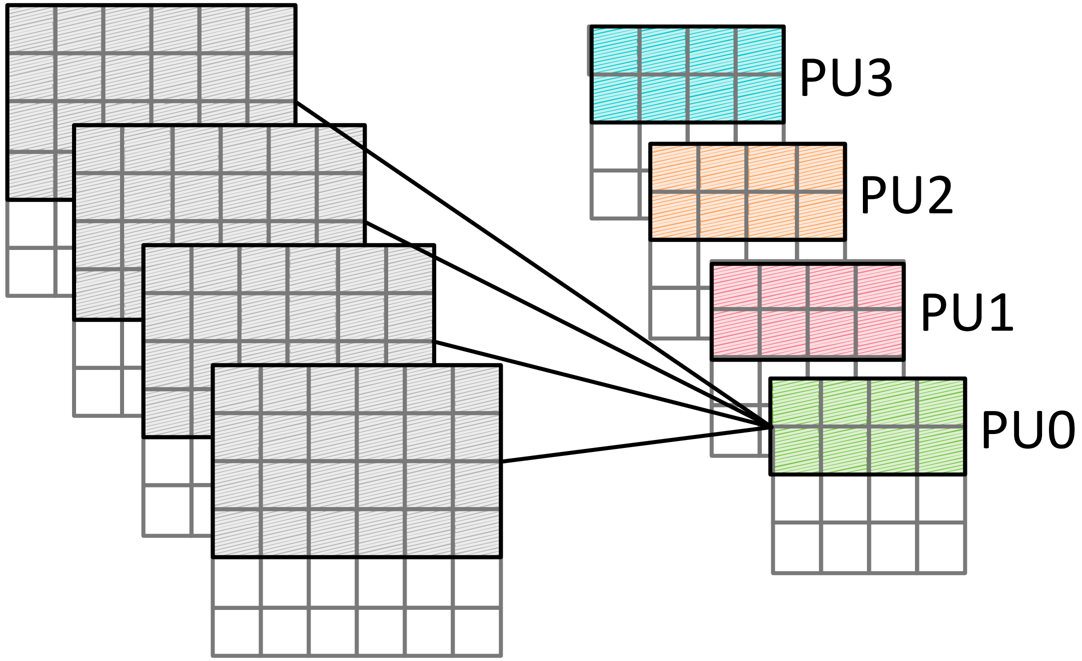

Normalization


Activation

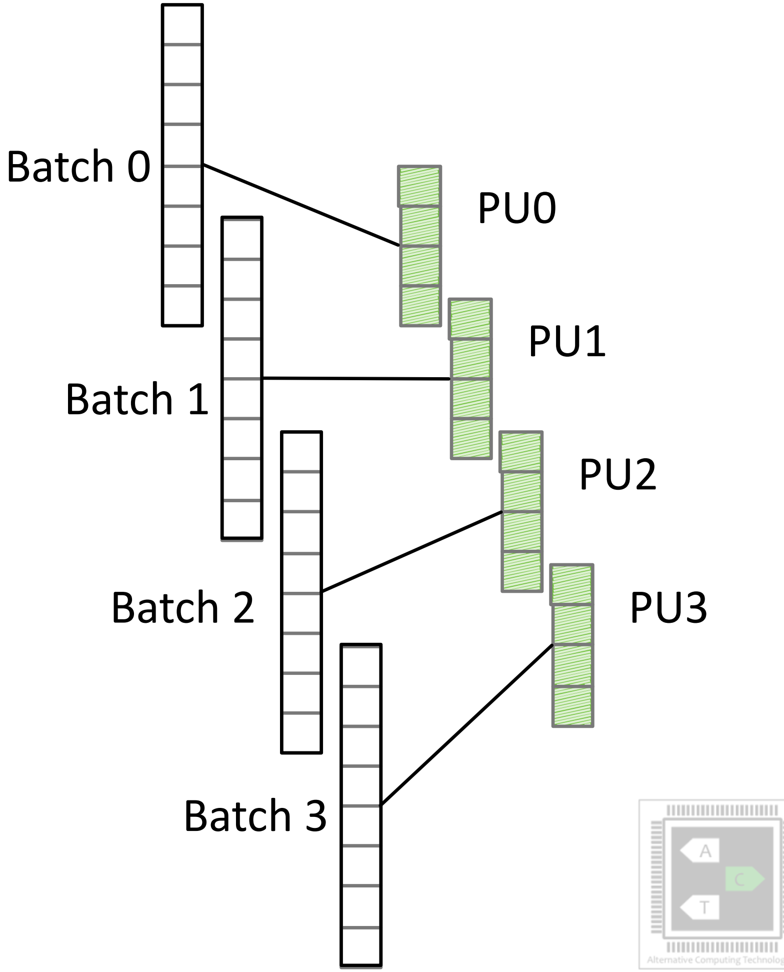
DnnWeaver supports all these five layers.

Scheduling Operations on Hardware

Convolution



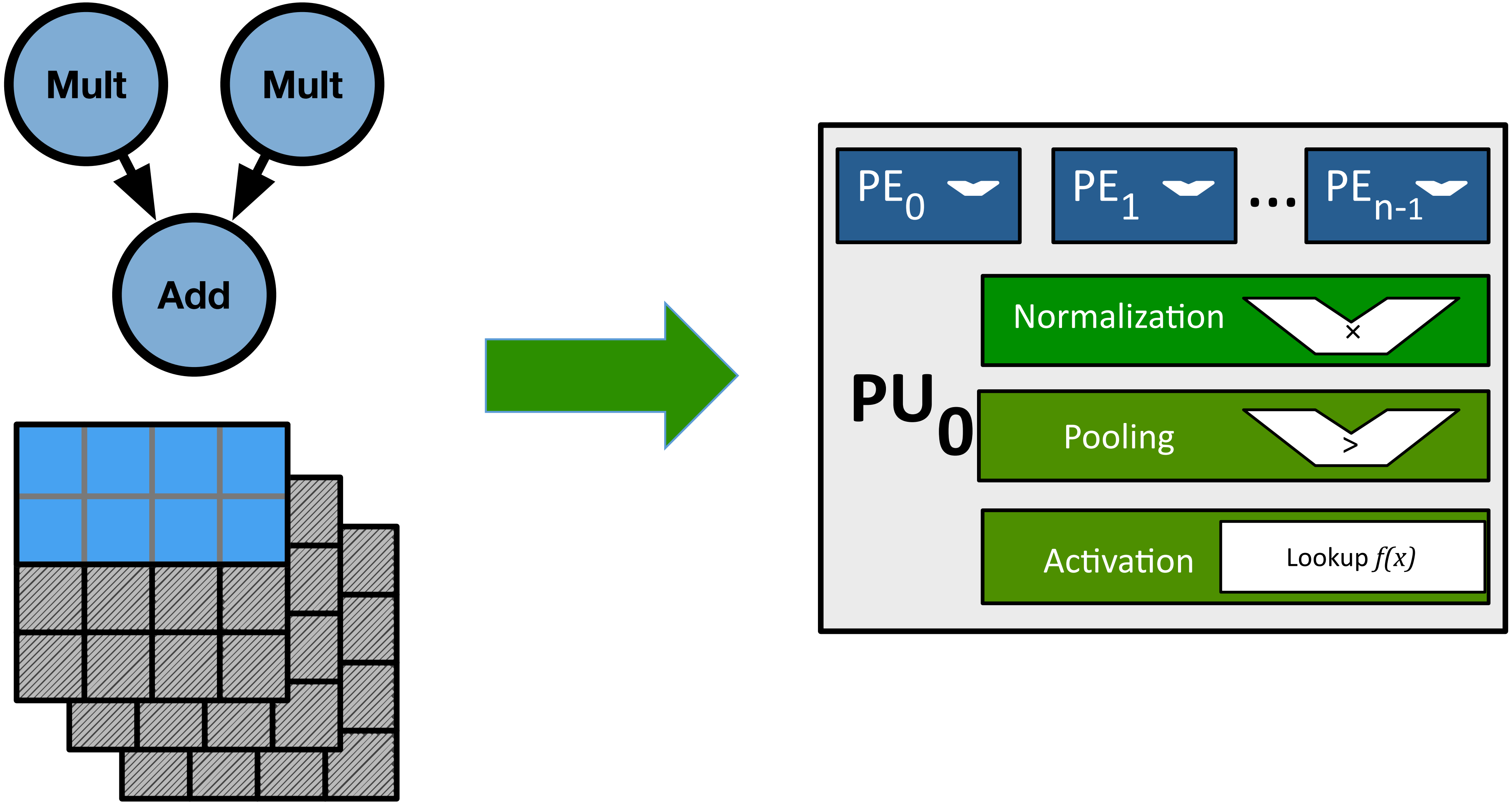
Inner-product



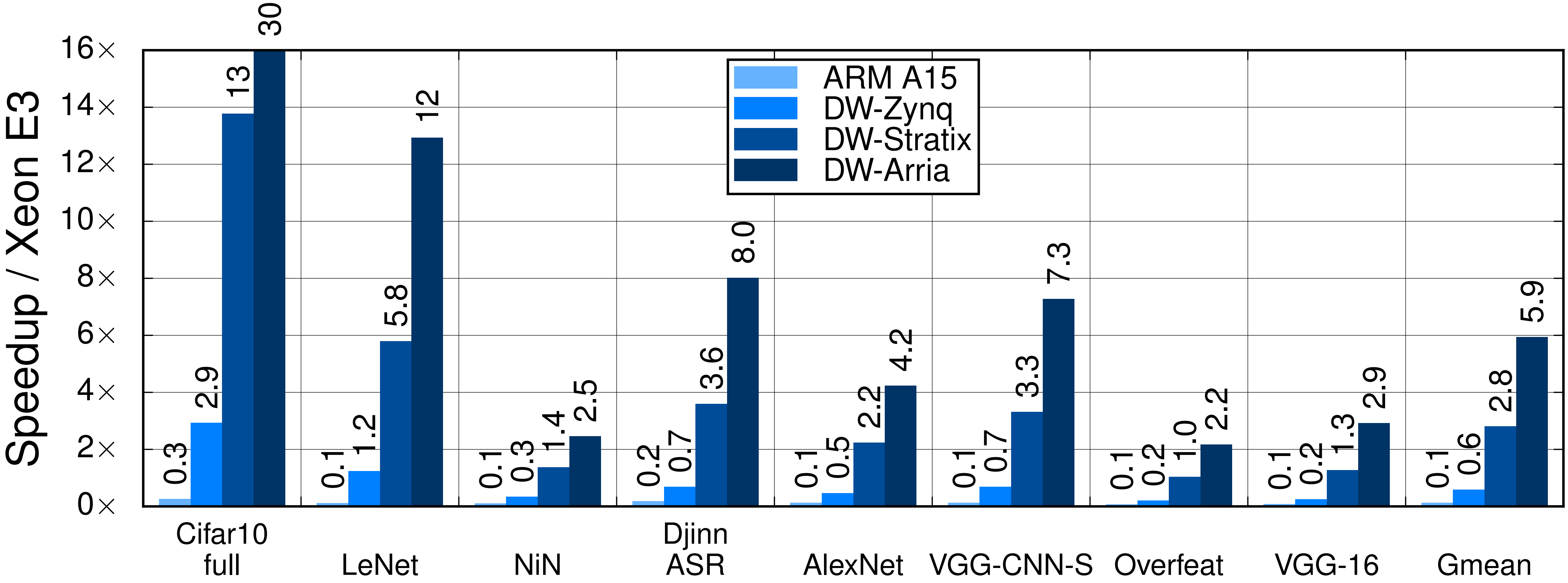
Performance vs GPU

Performance-per-Watt vs CPU

Executing the Slice

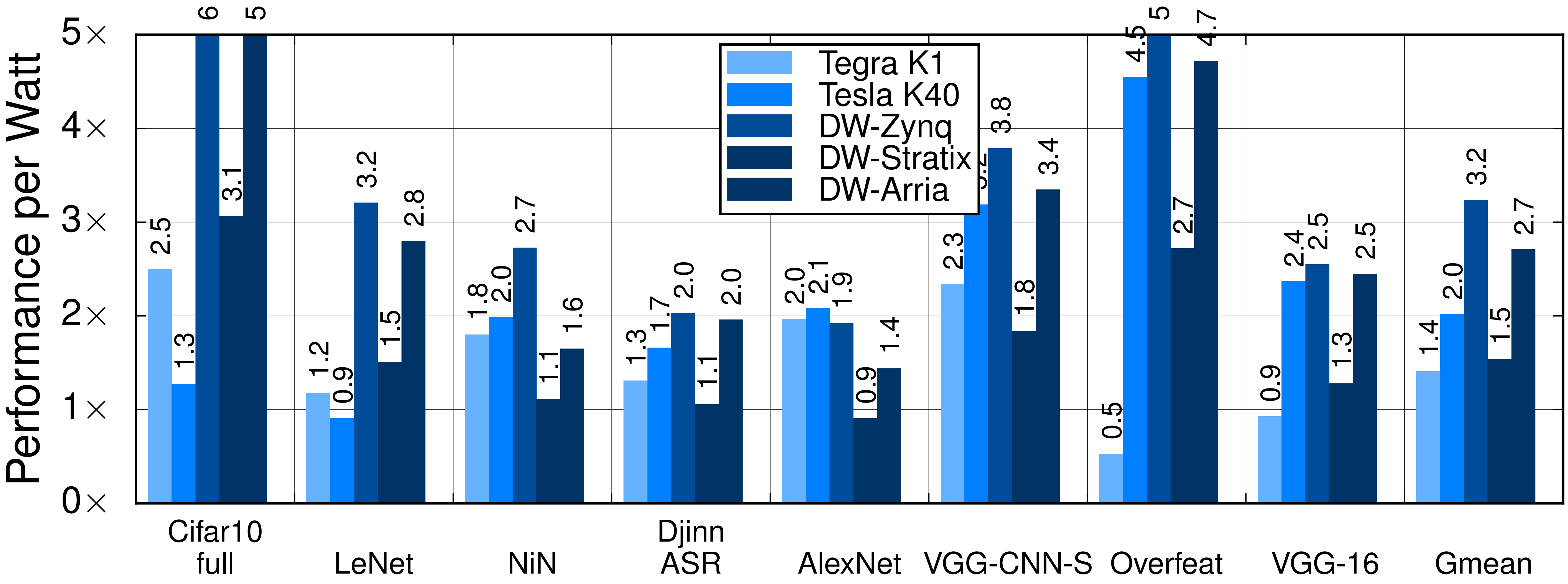


Performance vs CPUs



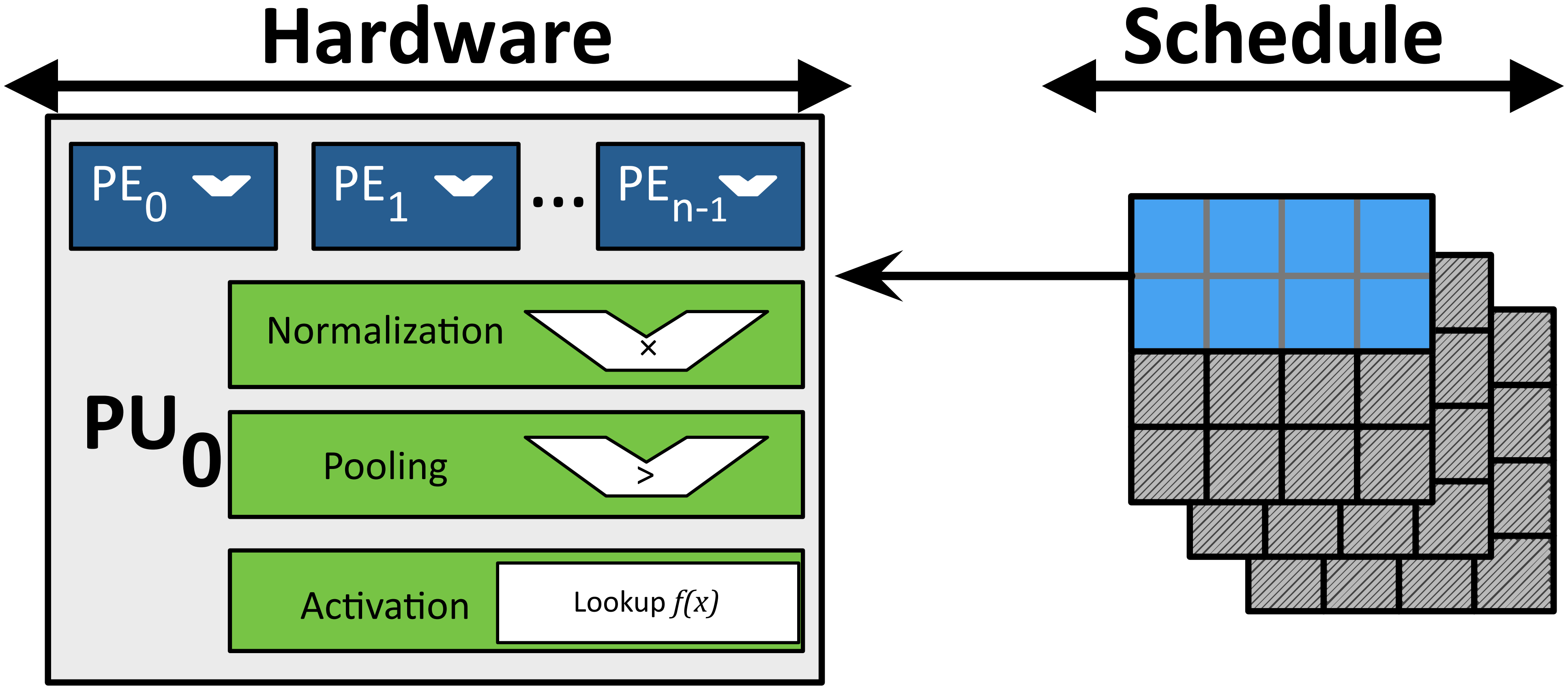
Compared to Xeon, Arria10 is **5.9x** faster, Stratix V is **2.8x** faster, and Zynq is **0.6x** faster.

Performance-per-Watt vs GPUs

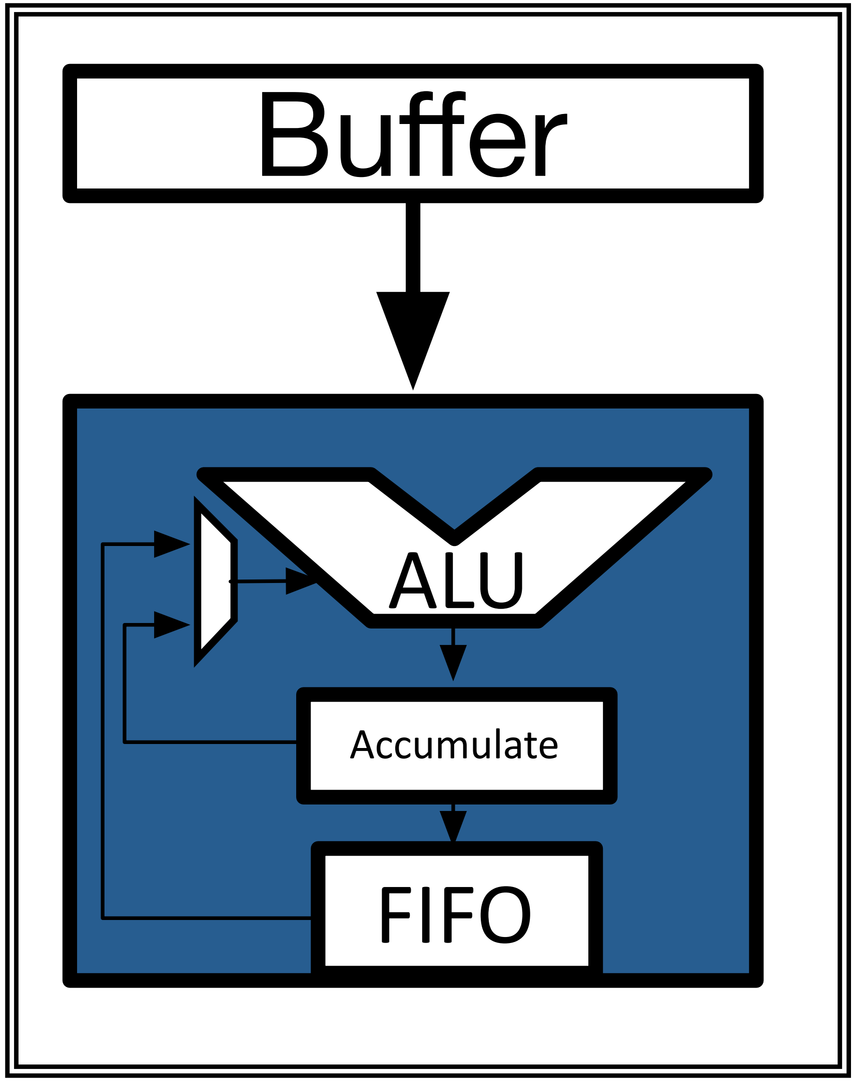


Compared to GTX650, Zynq is **3.2x**, Stratix V is **1.5x**, and Arria10 is **2.7x** more power efficient.

Co-optimize Hardware and Execution Schedule



Processing Engine


Buffer
ALU
Accumulate
FIFO