

Highly efficient programming environment for handling AI workloads

Tom Michiels, System Architect
Synopsys ARC[®] Processor Summit 2022



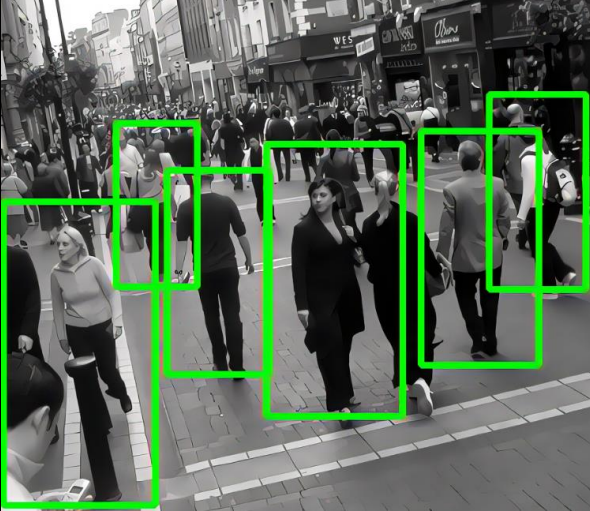
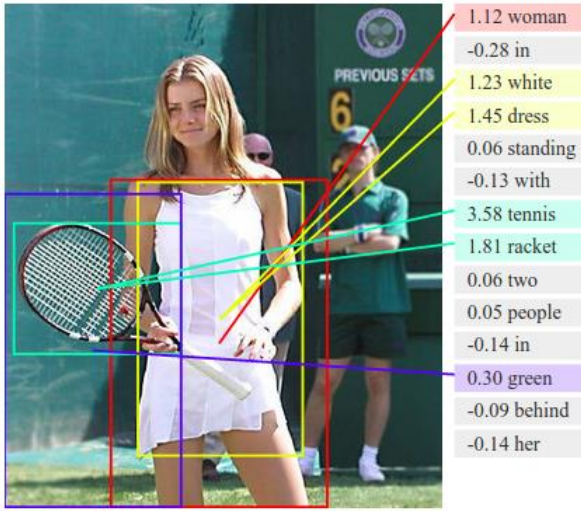

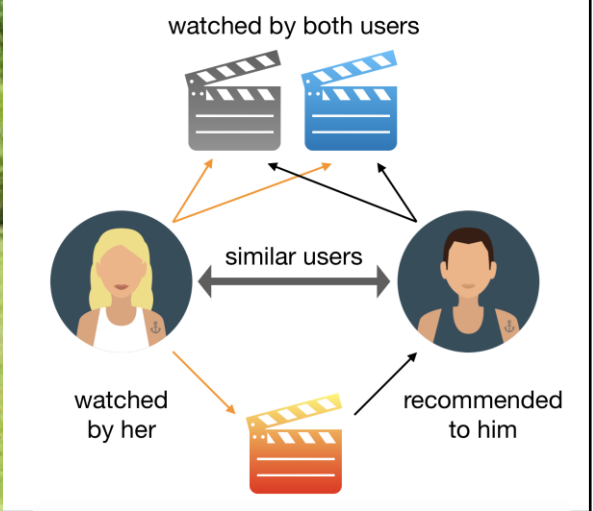
Agenda

- The AI Programming Challenge
- Optimizations For Programming AI-Enabled SoCs
- Quantifying The Benefits

The AI Programing Challenge



Popular & Emerging Neural Networks Are Still Evolving

CNN	RNN/LSTM	Transformers	Recommenders
	 <ul style="list-style-type: none"> 1.12 woman -0.28 in 1.23 white 1.45 dress 0.06 standing -0.13 with 3.58 tennis 1.81 racket 0.06 two 0.05 people -0.14 in 0.30 green -0.09 behind -0.14 her 	 <p>A woman throwing a frisbee in the park</p>	 <p>watched by both users</p> <p>similar users</p> <p>watched by her</p> <p>recommended to him</p>
<p>Vision, Lidar, Audio, Speech</p>	<p>Speech, Audio, Action Recognition</p>	<p>NLP, Speech, Vision</p>	<p>Commerce, Recommendations</p>

Convolutional Neural Networks process uncompressed images

Recurrent Neural Networks process sequential data like audio or speech streams

Uses parallelism and focused attention on relevant portions of image

Recommender system predicts future preference of a set of items for a user

Must “future-proof” your software to handle new ML graphs

AI Software Runs On a Spectrum Of Hardware Types

CPU, GPU, DSPs, NPUs, AI Accelerators...

Hardware	Performance	Area Efficiency	Power Efficiency	Flexibility	Typical Programming Model
CPU	★	★	★	★★★★★	C/C++ code
GPU	★★★★★	★	★	★★★★★	OpenCL or CUDA
FPGA	★★	★★★	★	★★★	Vendor Specific
DSP	★★★	★★★	★★★	★★★	C/C++ or OpenCL C
NPU	★★★★★	★★★★★	★★★★★	★★★	Vendor Specific
Accelerator	★★★★	★★★★★	★★★★★	★★	Hardwired or Special SDK

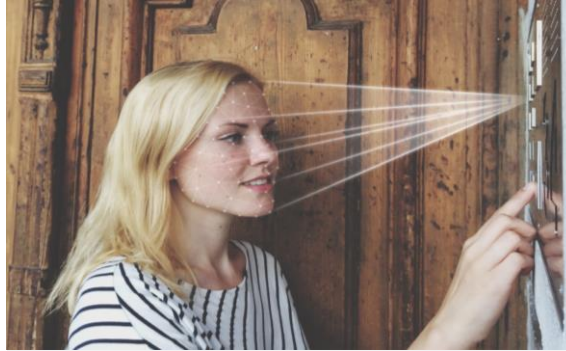
Ideally, your NN's will take advantage of any AI-enabled hardware

Wide Variety Of Performance For AI Edge Devices



- AIoT
- Human activity recognition

<100 GOPS



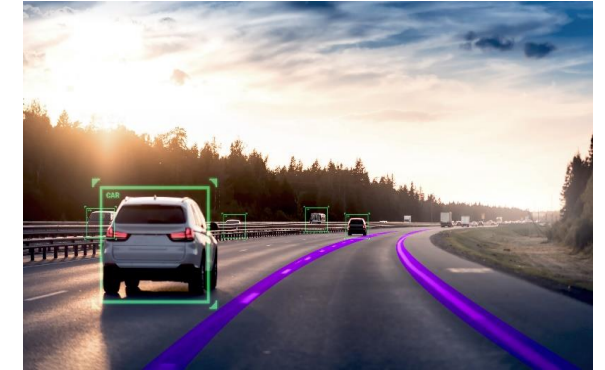
- Robotics / Drones
- Automotive Powertrain
- Games/toys
- Audio / Voice control
- Facial detection

100 GOPS to 1 TOPS



- Driver monitoring system
- Surveillance
- Facial recognition
- Digital still cameras
- High End Gaming
- Augmented reality
- Mid-end smartphones
- Facial recognition

1 to 10 TOPS

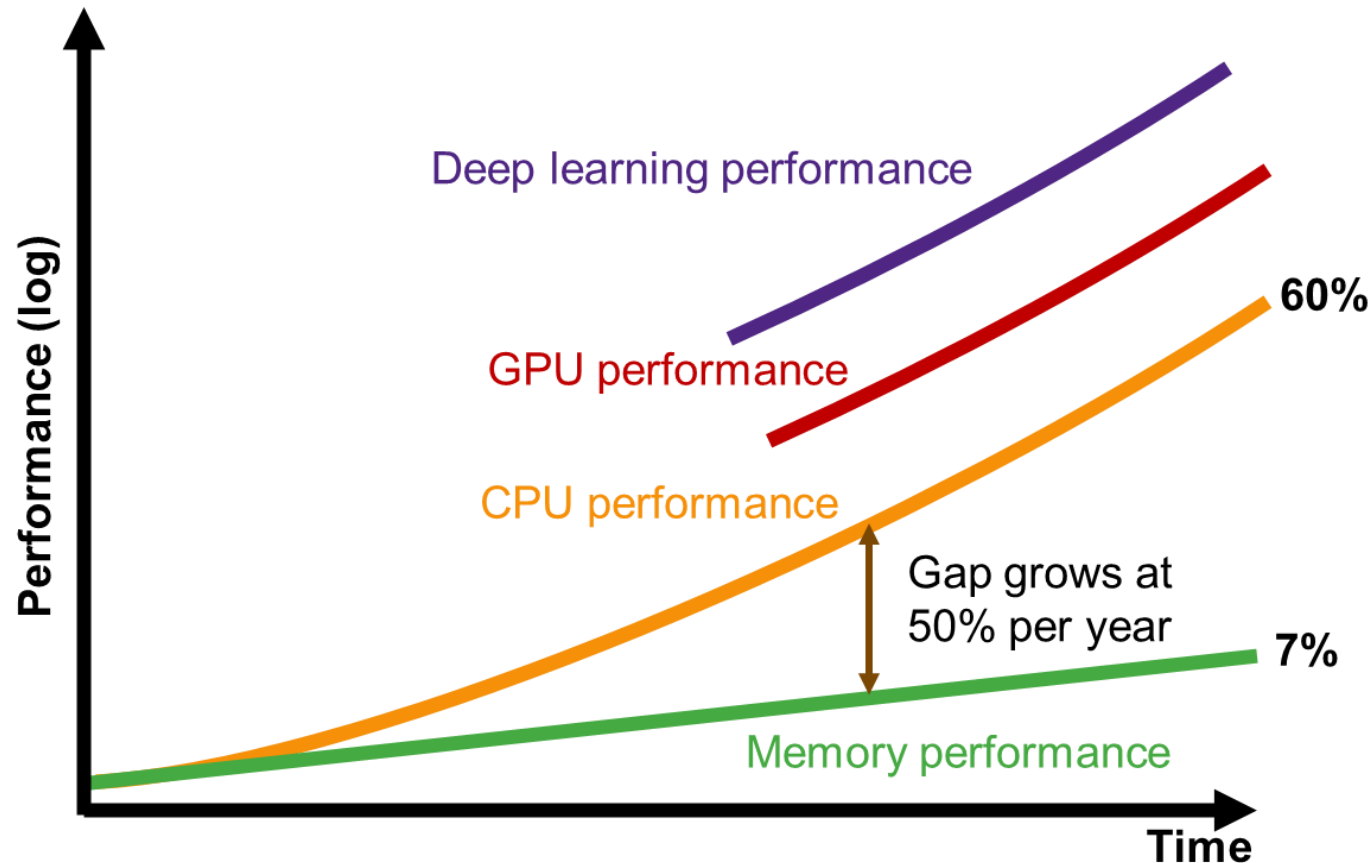


- ADAS Front Cameras
- ADAS LiDAR/Radar
- High end surveillance
- High-end smartphones
- DTV
- HPC
- Microservers (inference)
- Data center (inference)

10 to 1000+ TOPS

Same programming environment to serve multiple domains

Deep Learning Performance Outpacing Memory



- Moore's Law: CPU performance outpacing memory access speed
- GPUs initiated Deep Learning in 2012, widening the gap
- Deep Learning accelerators outpacing GPUs
- Goal: reduce data movement
 - Innovative heterogeneous memory architectures required
 - From on-chip memory compilers to high bandwidth HBM2

Limited memory bandwidth requires optimized data movements

Competing Machine Learning Frameworks

Lack of Programming Model Standardization for AI Algorithms

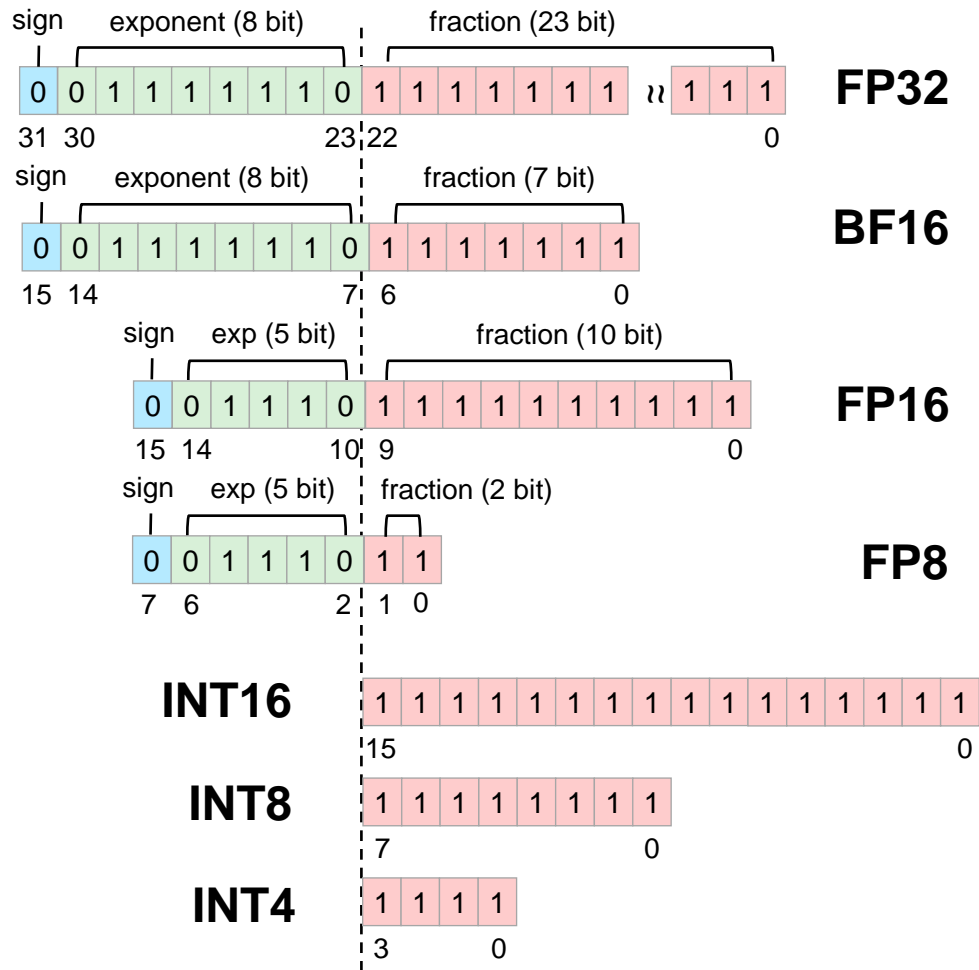


Programming model should support all popular frameworks

5 Optimizations For Programming AI-Enabled SOCs

1. Quantization
2. Multi-level Layer Fusion and Multi-level Tiling
3. Feature Map Compression/Decompression
4. Structured Sparsity
5. Featuremap partitioning

NN Applications Use Wide Range Of Data Representations



- **FP32** typical format used in GPUs for NN model training
- **FP16 & BF16** are NOT needed for accuracy over INT8/16 – they make the transition from GPU easier, avoids having to retrain models
- **FP8** has more traction for training than inference
- **INT16** provides accuracy ‘insurance’ for radar and super resolution (at reduced performance)
- **INT8** standard for neural network object detection
- **INT4** can save bandwidth; not very popular yet

Mixed Precision Quantization Enables Optimized Accuracy with Minimum Bandwidth Impact

Layer 1
8bit/8bit

Layer 2
8bit/8bit

Layer 3
8bit/8bit

Layer 4
8bit/8bit



Layer 1
8bit/8bit

Layer 2
16b/8bit

Layer 3
8bit/8bit

Layer 4
8bit/8bit



Optimize Accuracy with Minimum Bandwidth

Initial 8bit Quantized Model

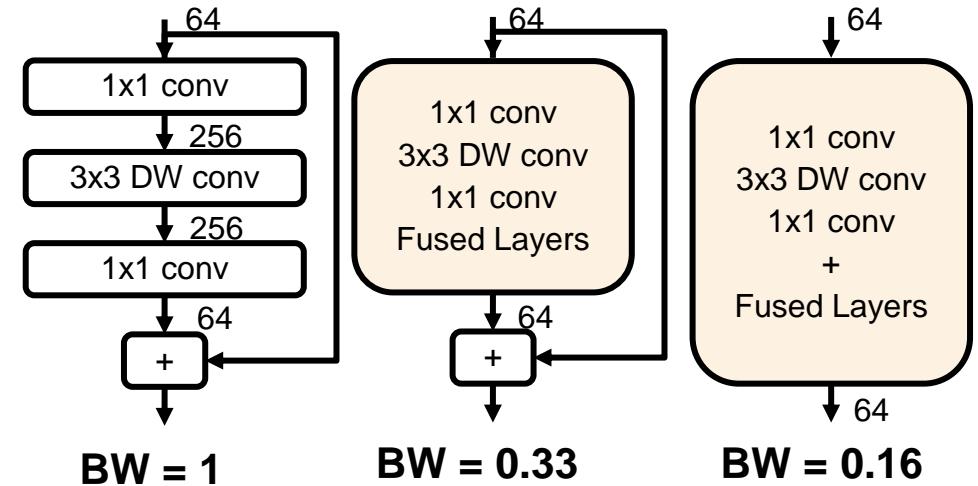
Mixed-Precision Quantized Model

B	C	D	E	F	G
run_config	maps_format	evgencnn_options_name	voc2012	voc2007	integral
bandwidth_110		toronto	17.1561403869	20.8071366317	17.0699625275
bandwidth_115		toronto	17.1428008212	20.7655613773	17.0589021067
bandwidth_120		toronto	17.0892989615	20.7337699279	17.011950925
bandwidth_130		toronto	17.0882782044	20.7275873169	17.0115620859
bandwidth_140		toronto	17.0741138523	20.7168843624	16.9925967794
bandwidth_145		toronto	17.1455956543	20.7264909609	17.06240228
bandwidth_150		toronto	17.0934892817	20.7061805589	17.0112040751
normal	12bit	toronto	17.0934892817	20.7061805589	17.0112040751
normal	8bit	toronto	0.1564007325	9.1005020715	0.1555868996

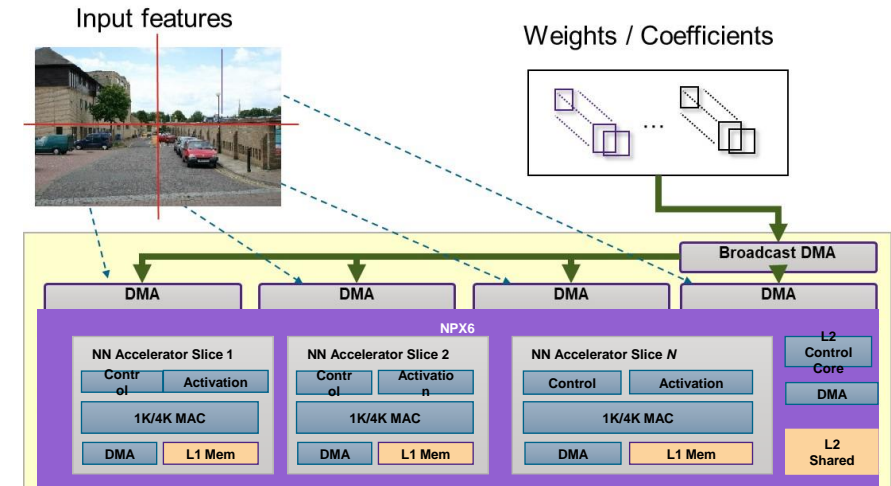
Techniques for Minimizes Bandwidth Requirements

- Multi-level Layer Fusion
 - Merging multiple folded layers into single primitives reduces feature map bandwidth
 - Merged layers can be fused into layers groups and tiled, taking advantage of L1 and L2 memories
- Coefficient Pruning and Compression
 - Coefficients with a zero value are skipped/counted, a compressed coefficient bitstream is created offline
 - Compression ratio can be increased through pruning and retraining
- Feature Map Compression
 - Lossless runtime compression and decompression of feature maps to external memory
 - Approx. 40% feature-map bandwidth reduction, exploiting sparsity
- Layer, Frame based and Feature Map Partitioning with DMA Broadcasting
 - Broadcast of common data across slices to minimize bandwidth of coefficients and feature-maps loading

Multi-level Layer Fusion MobileNet v1/v2

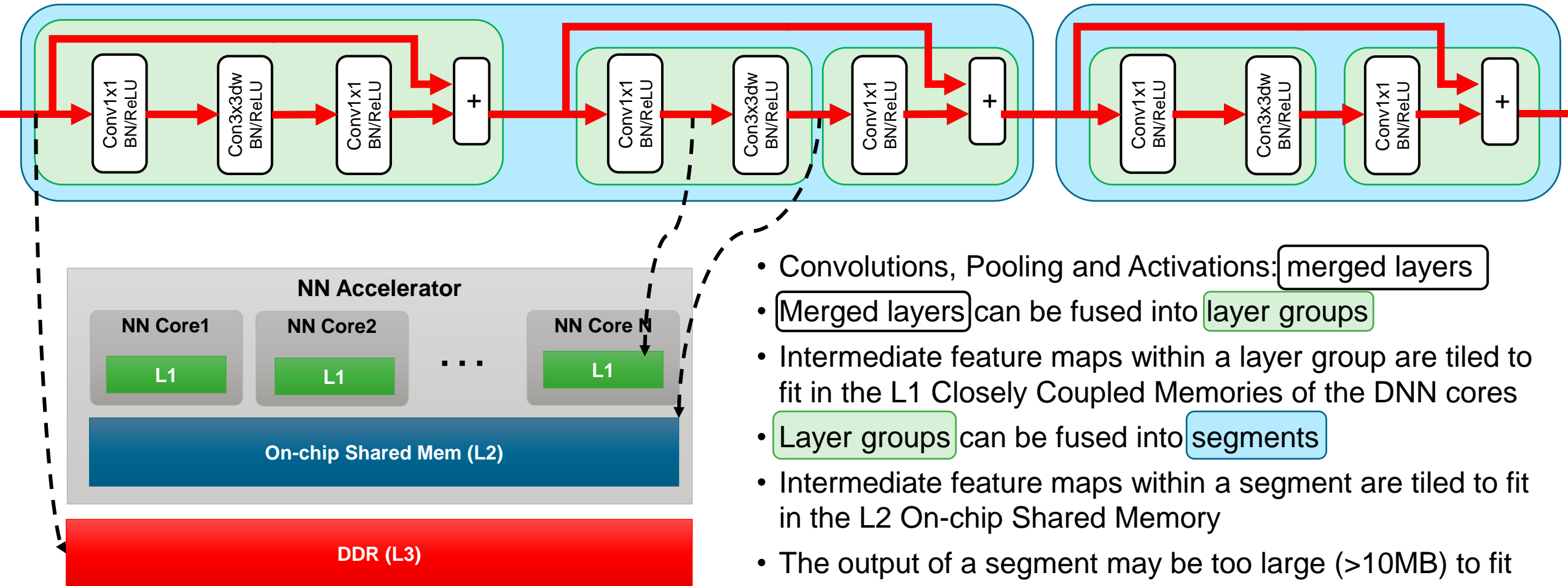


Feature Map Partitioning / DMA Broadcasting



Advanced Data Bandwidth Reduction Techniques

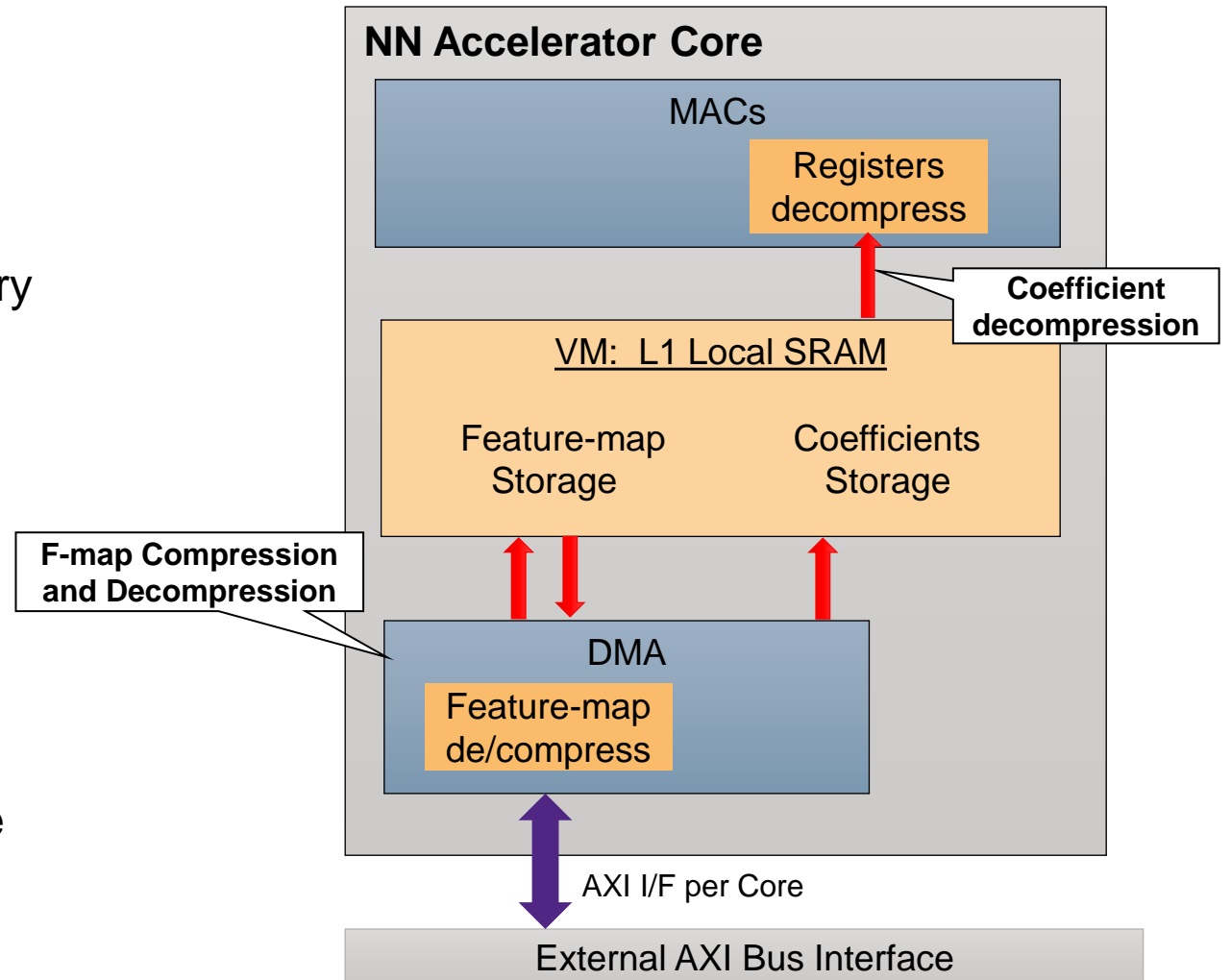
Multi-level Layer Fusion and Multi-level Tiling



- Convolutions, Pooling and Activations: merged layers
- Merged layers can be fused into layer groups
- Intermediate feature maps within a layer group are tiled to fit in the L1 Closely Coupled Memories of the DNN cores
- Layer groups can be fused into segments
- Intermediate feature maps within a segment are tiled to fit in the L2 On-chip Shared Memory
- The output of a segment may be too large (>10MB) to fit in L2 On-chip Shared Memory and is spilled to L3 DDR

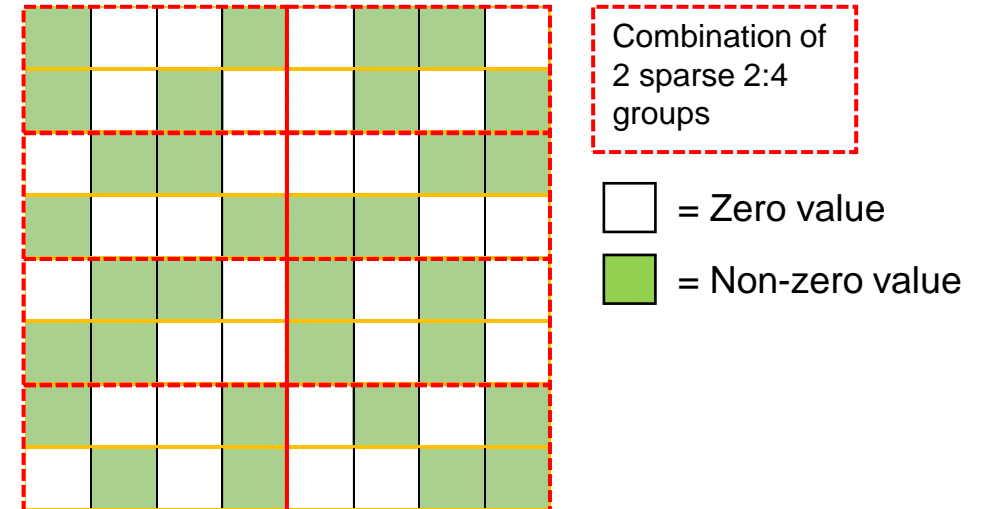
Data Compression/Decompression

- Coefficient Pruning
 - Coefficients with a zero value are skipped/counted
 - Decompression done between local VM memory and NN datapath registers
 - Offline coefficient pruning (with retraining) can increase proportion of zero coefficients
 - Support of structured and unstructured sparsity
- Feature map compression/decompression
 - Runtime compression and decompression
 - NN core DMA supports HW compression mode
 - Bandwidth reduction of 40~45% measured typically



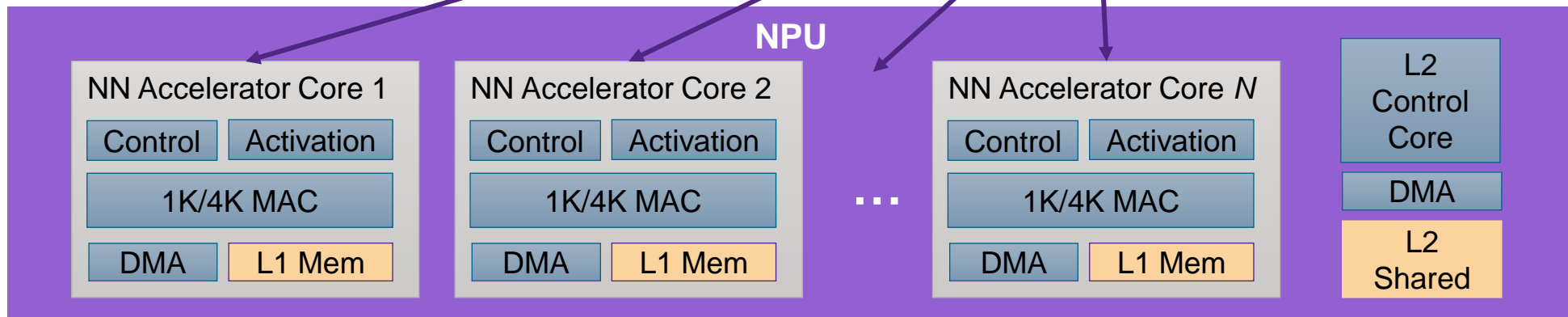
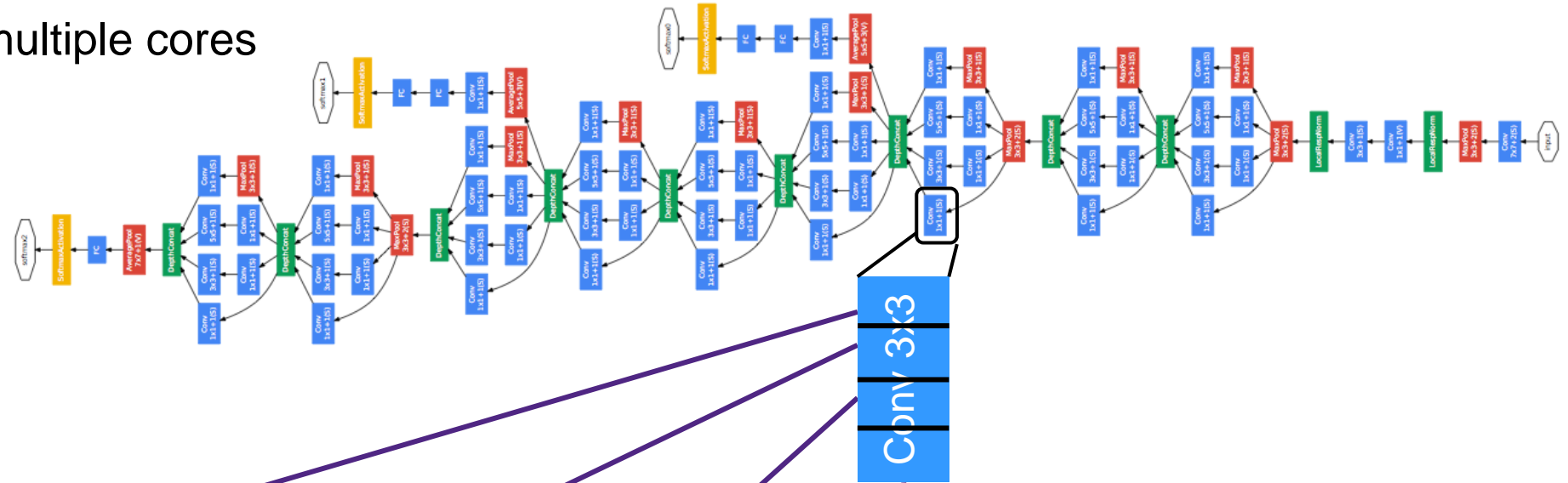
Structured Sparsity Can Improve Performance 2X

- Sparsity takes advantage of a matrix of numbers that includes many zeros or values that will not significantly impact a calculation
- Can exploits sparsity in coefficients
 - Flexible use of sparsity in coefficient vectors in channel dimension
 - Effective speedup of 1.4X~1.8X with almost no accuracy loss
- Doubles the effective MACs on applicable layers
- Requires pruning and retraining
 - No accuracy loss for key model families: e.g. ResNet, ResNext, Densenet, Bert, GNMT
 - Other models may have accuracy vs. performance tradeoffs



Latency Reduction via Feature-map partitioning

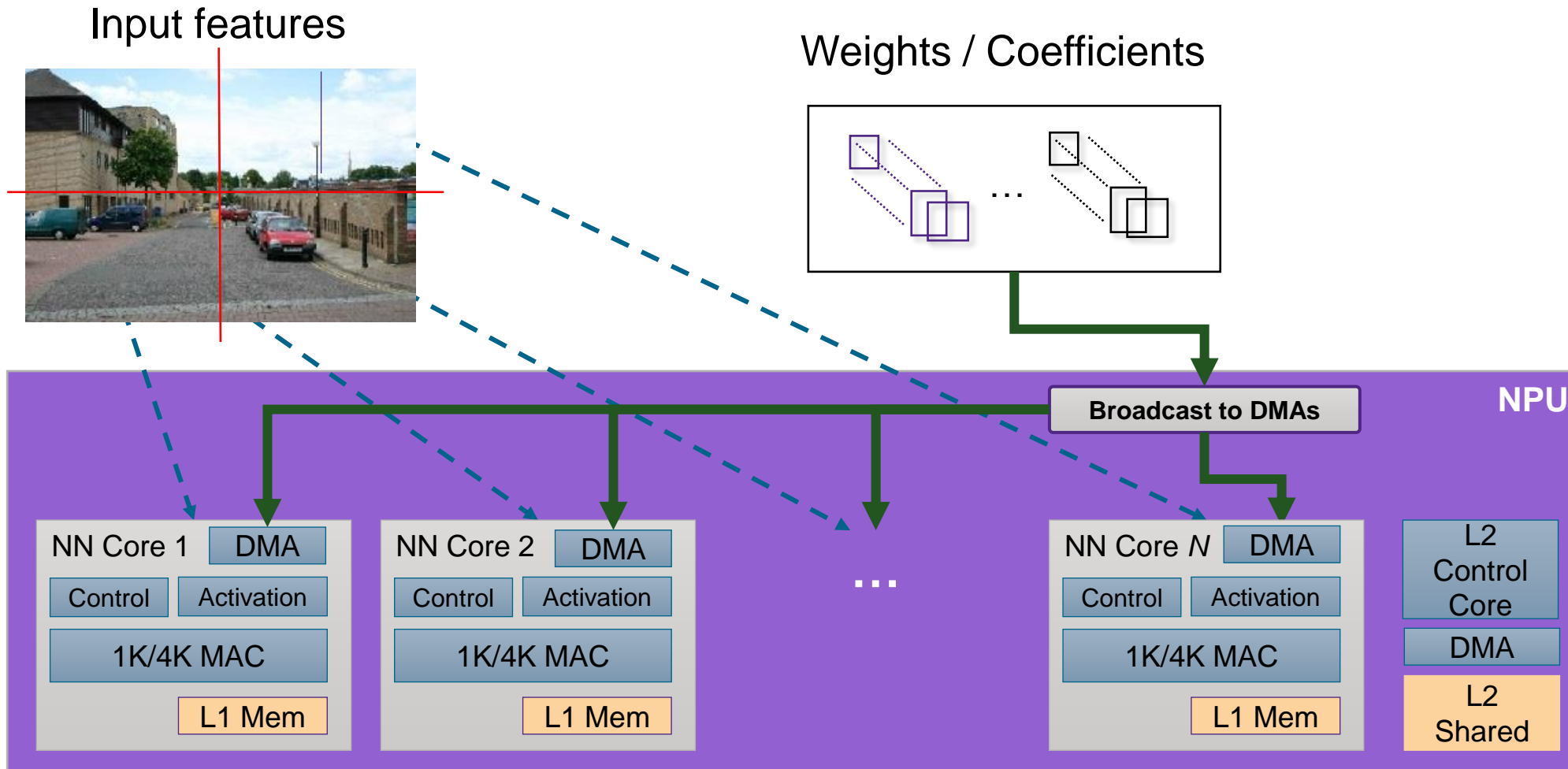
Split each layer over multiple cores



- Higher throughput – up to NX
- Lower latency – up to NX – due to parallel processing of a layer
- Significant bandwidth reduction (via DMA broadcasting)

Feature-map partitioning – contd.

Spatial partitioning: Reuse weights across cores through a broadcast DMA



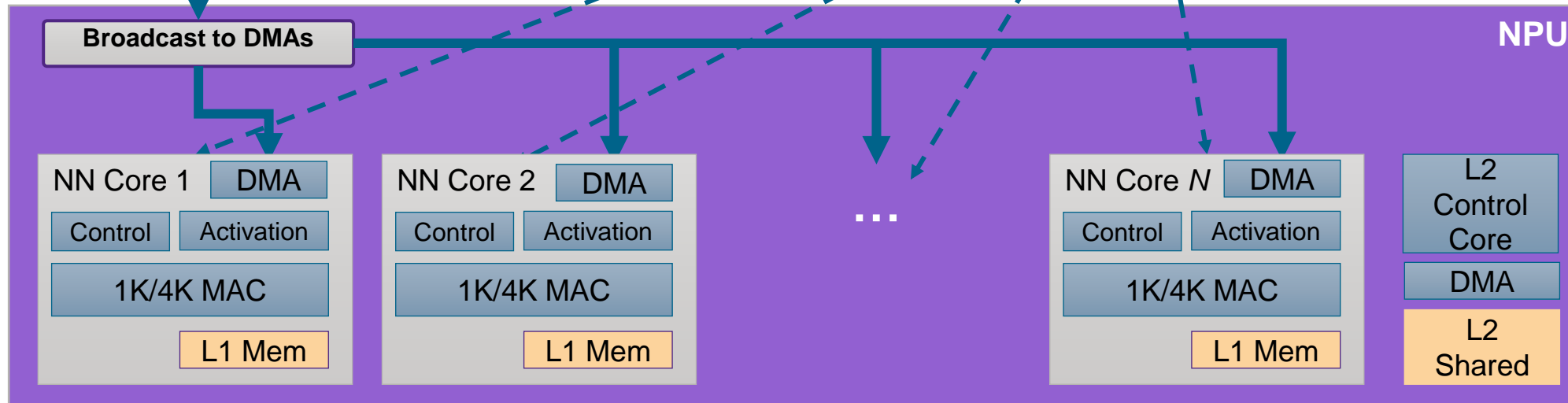
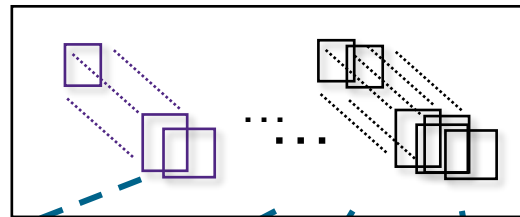
Feature-map partitioning – contd.

Channel partitioning: Reuse features across cores through a broadcast DMA

Input features



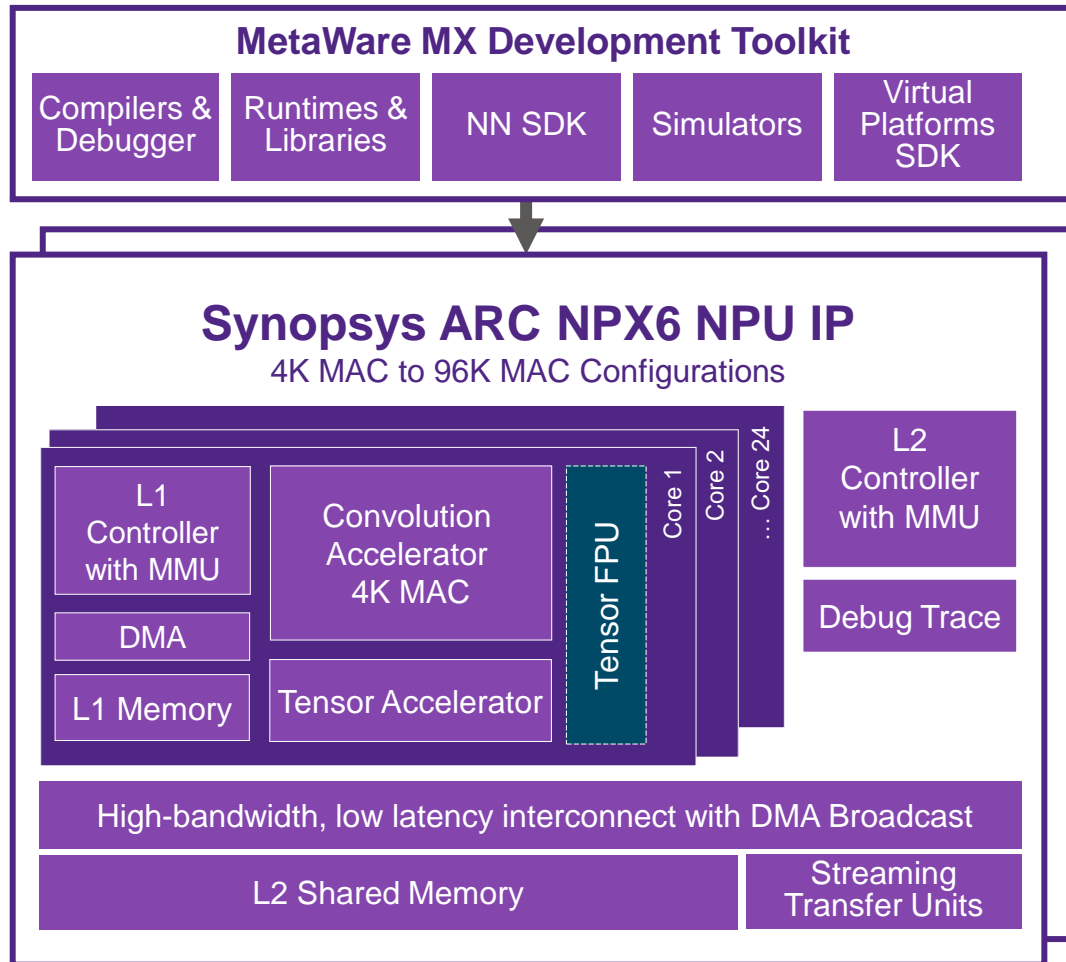
Weights / Coefficients



Quantifying the Benefits



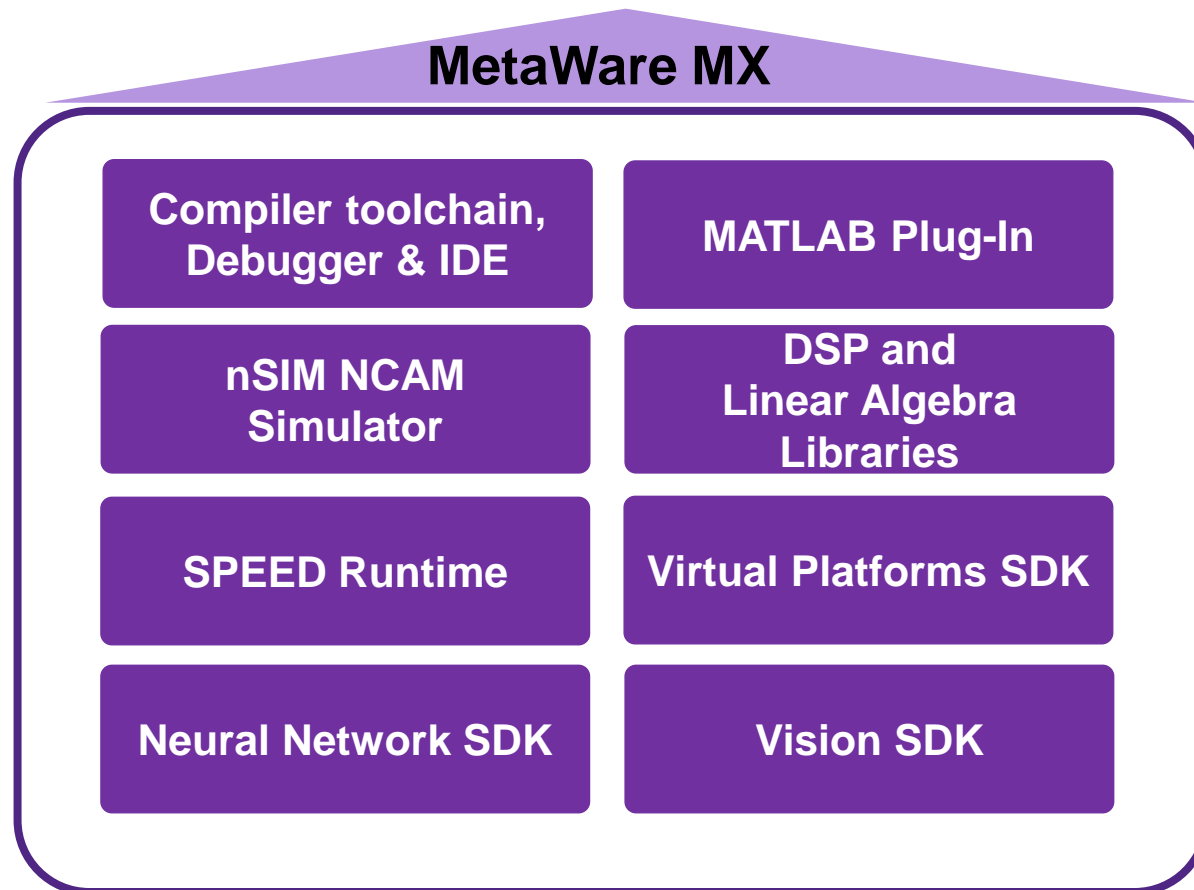
Synopsys Introduces ARC NPX6 NPU and MetaWare MX



- Scalable NPX6 architecture
 - 1 to 24 core NPU up to 96K MACS (440 TOPS*)
 - Multi-NPU support (up to eight for 3500 TOPS*)
- Trusted software tools scale with the architecture
- Convolution accelerator – MAC utilization improvements with emphasis on modern network structures
- Generic Tensor accelerator – Flexible Activation & support of Tensor Operator Set Architecture (TOSA)
- Memory Hierarchy – high bandwidth L1 and L2 memories
- DMA broadcast lowers external memory bandwidth requirements and improves latency

* 1.3 GHz, 5nm FFC worst case conditions using sparse EDSR model

Modular Toolkit Supports Control, DSP, Vision and ML Software Development



DesignWare® ARC® MetaWare MX Development Toolkit

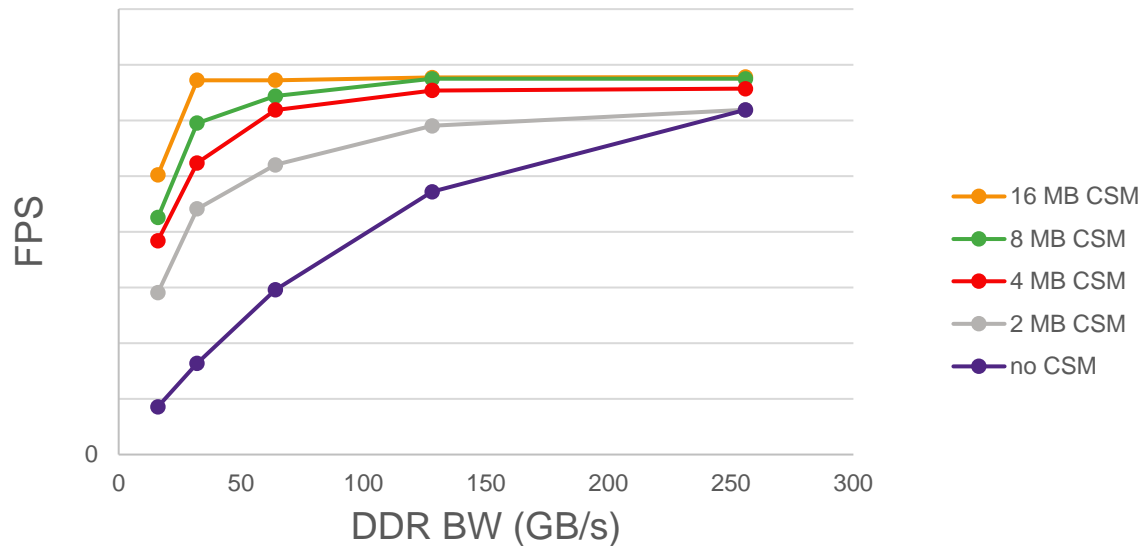
- Integrated toolkit provides optimizing compilers, debugger, libraries and a simulator for development on ARC processors
- Includes Vector DSP and Linear Algebra Libraries (BLAS/LAPACK) and MATLAB Plug-In for Model-Based Design Environment
- MetaWare Neural Network SDK for enabling and optimizing Machine Learning and inference applications
- Includes simulation platforms for early software development and architectural exploration with MetaWare Virtual Platforms SDK
- Development of Computer Vision for pre- & post-processing eased with MetaWare Vision SDK

Benchmark Performance vs. L2 CSM size and DDR Bandwidth

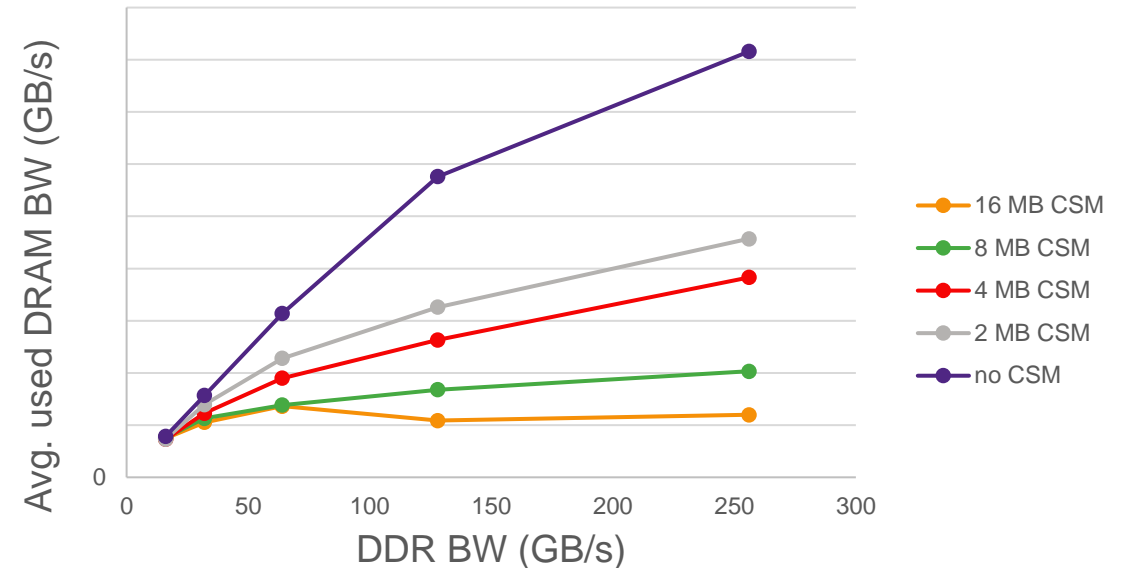
Result for selected NPX6-32K config – without structured sparsity

- NPX6 configuration: 8 NN cores * 4096 MACs per core
- NN core internal memory (L1): 384 KB per NN core
- Cluster Shared Memory (L2): 0 to 16 MB
- Ext. DRAM bandwidth (L3): 16, 32, 64, 128, 256 GB/s
- 8 bit data

scaled_yolo5(960x544) on NPU32K(384 KB)



scaled_yolo5(960x544) on NPU32K(384 KB)



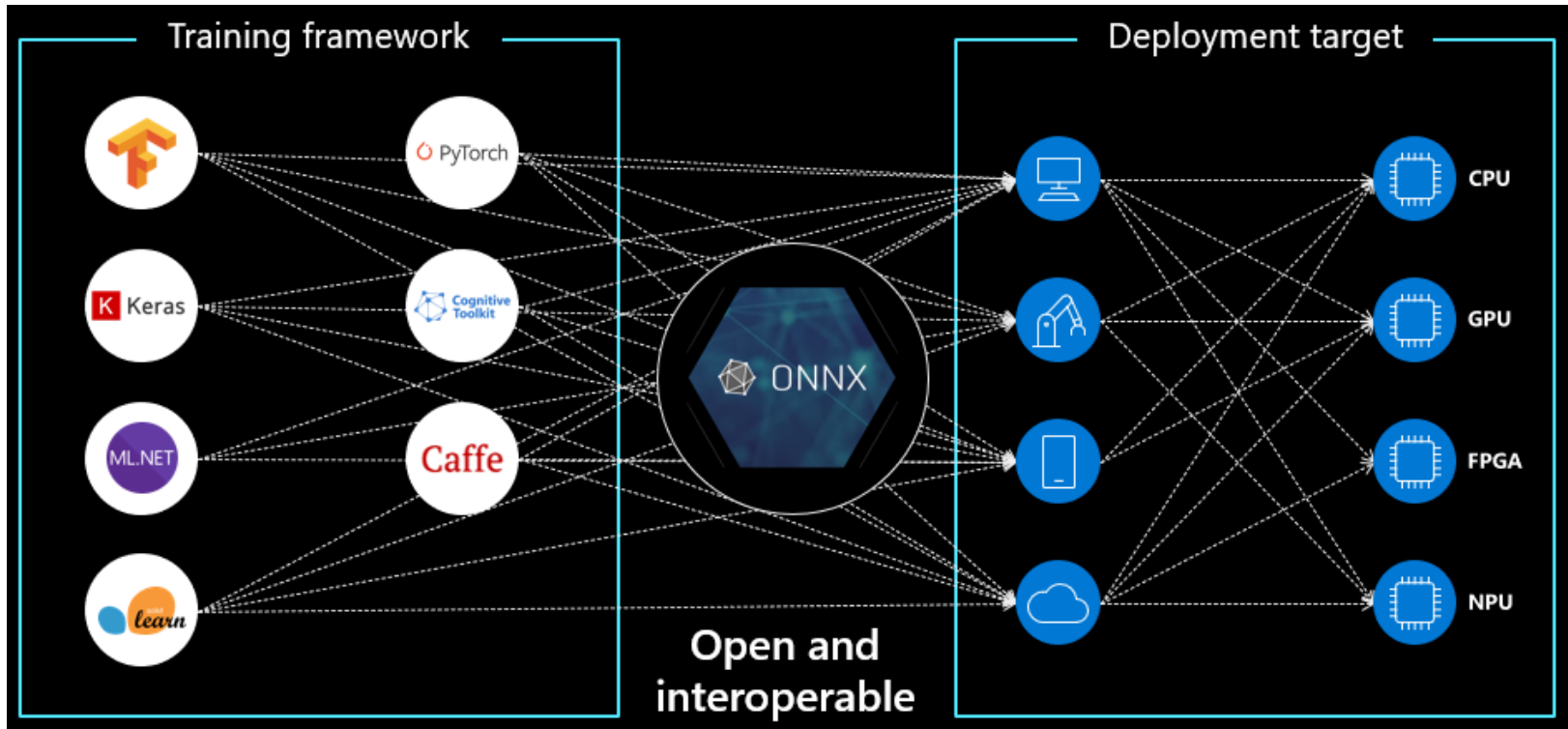
Performance Gains Obtained with Structured Sparsity

	NPX6-4K	NPX6-16K	NPX6-64K
Graph	% FPS improvement With Structured Sparsity	% FPS improvement With Structured Sparsity	% FPS improvement With Structured Sparsity
Inception v3	151%	142%	124%
Inception v3 FHD	148%	148%	148%
ResNet-50 v1.5	146%	147%	128%
ResNet-50 v1.5 FHD	142%	147%	147%
MobileNet v2	124%	133%	114%
MobileNet v2 FHD	120%	121%	117%
Yolo v3	152%	171%	165%
Yolo v3 FHD	165%	164%	168%
SSD-ResNet34	167%	171%	171%
SSD-MobileNet	151%	138%	115%
DeepLab v3	127%	129%	128%
EDSR	200%	191%	190%
SRGAN	176%	173%	171%
BERT_large	128%	135%	147%
BERT_large (batch=4)	128%	163%	166%
Vit_B_16	144%	128%	154%
Vit_L_16	132%	145%	149%
Vit_H_16	129%	145%	144%
swin_tiny	148%	148%	134%
swin_small	156%	158%	136%
swin_base	153%	163%	143%

ONNX to the Rescue

Open Neural Network Exchange

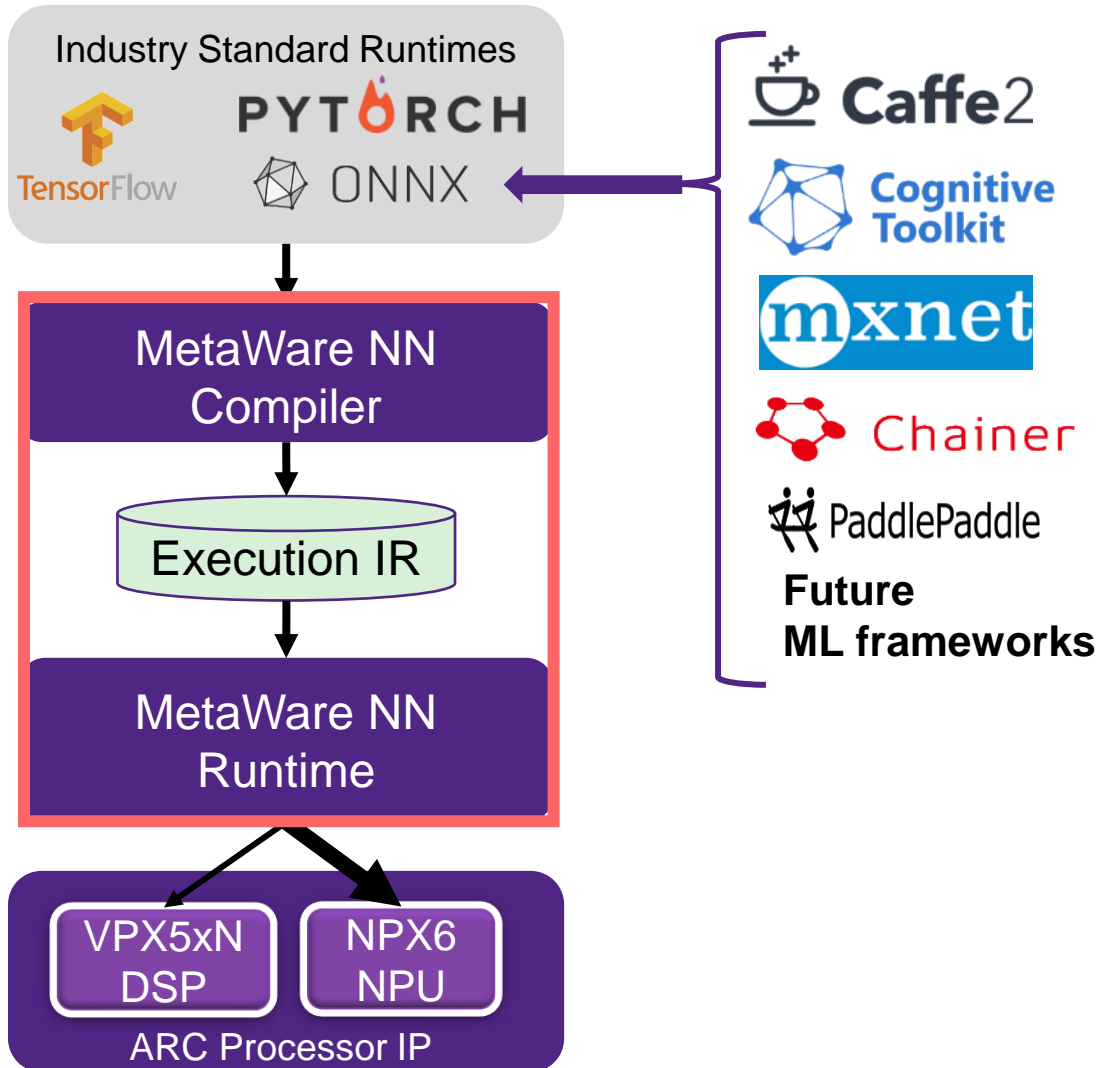
The open standard for machine learning interoperability



- Helps solve the challenge of hardware dependency related to AI models
- Open format to represent both deep learning and traditional models
- Defines a common set of operators and file format
- AI developers can use models with a variety of frameworks, tools, runtimes, and compilers
- Enables deploying same AI models to multiple HW-accelerated targets

Source: Microsoft

Support for Different Programming Frameworks



- MetaWare NN Compiler integrates with standard frameworks
- Automatic mapping to NPX6 and VPX5 vector DSP with no manual optimization required
 - User-driven optimization options:
e.g. Latency, throughput, bandwidth
- Generated code can run on multiple development platforms
 - Fast Performance Models (FPM)
 - Zebu H/W Emulator
 - HAPS FPGA board

State-Of-The-Art System Level Modeling And Analysis

Architecture Design

- **Fast Performance Model**

- Fast cycle-based Performance Model of NPX6 (and VPX5 cores)
- Integrated Platform Architect simulation environments

Software Development

- **Virtualizer Virtual Prototyping**

- VDK (Virtualizer development Kit for early Software Development Platform)

Power profiling

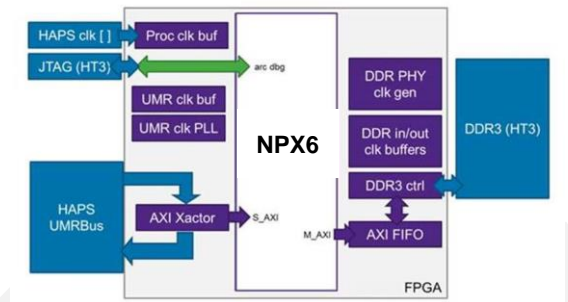
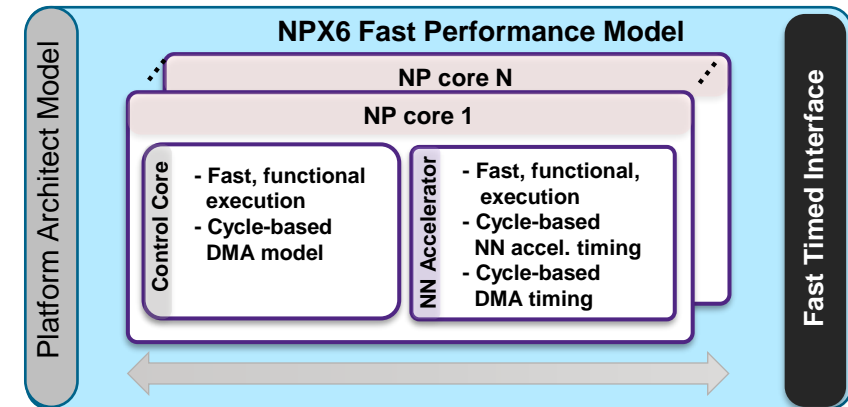
- **ZeBu Emulation**

- Accurate performance and power modeling

Benchmarking & Profiling

- **HAPS Prototyping**

- NPX6 mapped to HAPS board provides cycle accurate performance for benchmarking and software development



Summary

- AI Programming is a challenge amid evolving Neural Networks, absence of a standard programming model and the wide spectrum of HW types. A key challenge is the limited memory bandwidth
- Synopsys advanced optimizations for AI includes Mixed Precision Quantization to increase accuracy, Data Bandwidth Reduction techniques like multi-level tiling, Feature Map Partitioning to minimize bandwidth requirements, and Structured Sparsity utilization
- Synopsys MetaWare MX Development Toolkit supports different programming frameworks, different HW targets, is extensible, and includes state-of-the-art system level modeling

Thank You

