Pervasive and Sustainable AI with Adaptive Computing

Tryggve Mathiesen
AMD FAE Gothenburg Sweden

Michaela Blott, Thomas Preusser
AMD Research & Advanced Development

AMD
together we advance_

AMD Gothenburg – home of MicroBlaze V Team, Sweden – AECG-SSD

Göran

David

Stefan

Rikard

Roger

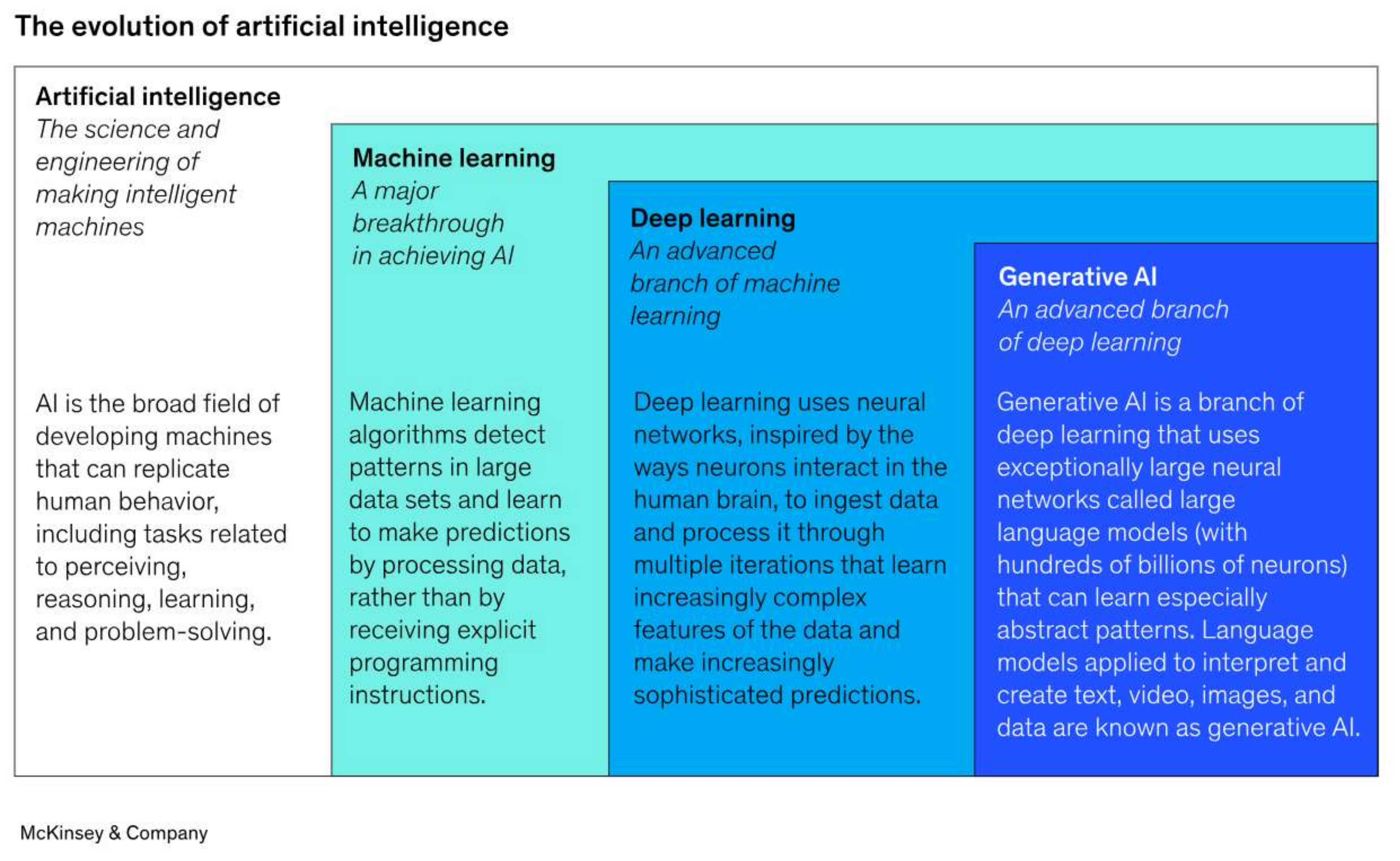
Center of Excellence for Processor Development and Cache Coherency

- Classic MicroBlaze
  - Enhanced versions approximately once a year since launch 2001
  - 10000+ soft IP customer instantiations every month
  - Hard MicroBlaze IP subsystems in Zynq MPSoC (3), Versal (14+), …
  - 32-bit and 64-bit proprietary ISA
  - Linux capable memory management
  - Triple Modular Redundancy/Lockstep
- MicroBlaze V (RISC-V – Open Specification Processor Architecture ISA)
  - First customer early access in Vivado 2023.2
  - Continued enhancements and development ongoing
  - Utilizes MicroBlaze code base
  - Aimed for both soft IP and hard IP in AMD devices
  - Plug-and-Play compatible with classic MicroBlaze
  - Enabling RISC-V Open Source SW community tools/flow
- System Cache
  - Accelerator Coherency (ACE, CCIX, CHI, CXL protocols)
  - L2 cache for MicroBlaze

AMD Research and Advanced Development (RAD)

- **Integrated Comms and AI Lab**
  - ~20 researchers plus university program
    - 5 different locations
  - Established as Xilinx Research Labs 18 years ago
- **Focus: AI and Communications**
  - Building systems, architectural exploration, algorithmic optimizations, benchmarking
  - In collaboration with partners, customers, and universities
    - ETH Zuerich, Paderborn University, Imperial College, KIT, NTNU, Politecnico di Milano, NUS, University of Sydney

Evolution of AI – Generation of Artificial intelligence



The evolution of artificial intelligence

Artificial intelligence
The science and engineering of making intelligent machines

AI is the broad field of developing machines that can replicate human behavior, including tasks related to perceiving, reasoning, learning, and problem-solving.

Machine learning
A major breakthrough in achieving AI

Machine learning algorithms detect patterns in large data sets and learn to make predictions by processing data, rather than by receiving explicit programming instructions.

Deep learning
An advanced branch of machine learning

Deep learning uses neural networks, inspired by the ways neurons interact in the human brain, to ingest data and process it through multiple iterations that learn increasingly complex features of the data and make increasingly sophisticated predictions.

Generative AI
An advanced branch of deep learning

Generative AI is a branch of deep learning that uses exceptionally large neural networks called large language models (with hundreds of billions of neurons) that can learn especially abstract patterns. Language models applied to interpret and create text, video, images, and data are known as generative AI.

McKinsey & Company

Example:

*Chess Computers
*Web Search
*Prediction future
*MR screening
*ADAS ....
*Art/Picture Gen
*Coding Support
*Verification Analyze
*Authoring
*Deep Fake/Frauds
....

“LLM + Deep Memory”
-Understand Text
-Understand speech
-Understand images
-Understand “life”?
-Adaptable:True/False?
-Evolve/mutate: DNA?

DNNs and Their Potential

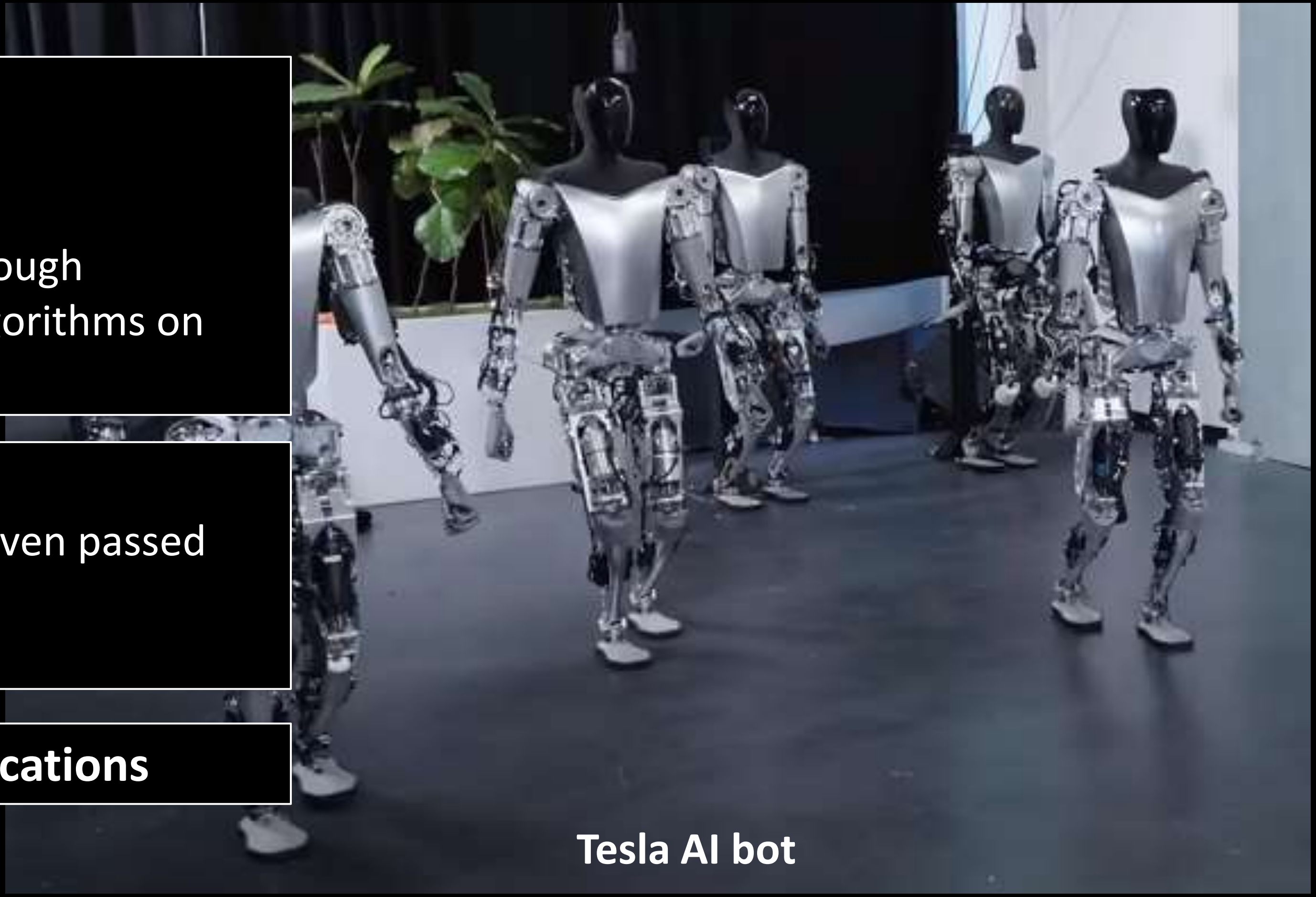
Huge potential

- Requires little domain expertise
- NNs are a “universal approximation function”
- If you make it big enough and train it long enough
  - Can outperform humans and existing algorithms on specific tasks

Solves previously unsolved problems

- Code, text and image generation, and GPT-4 even passed the bar exam in the 90th percentile
- Protein folding

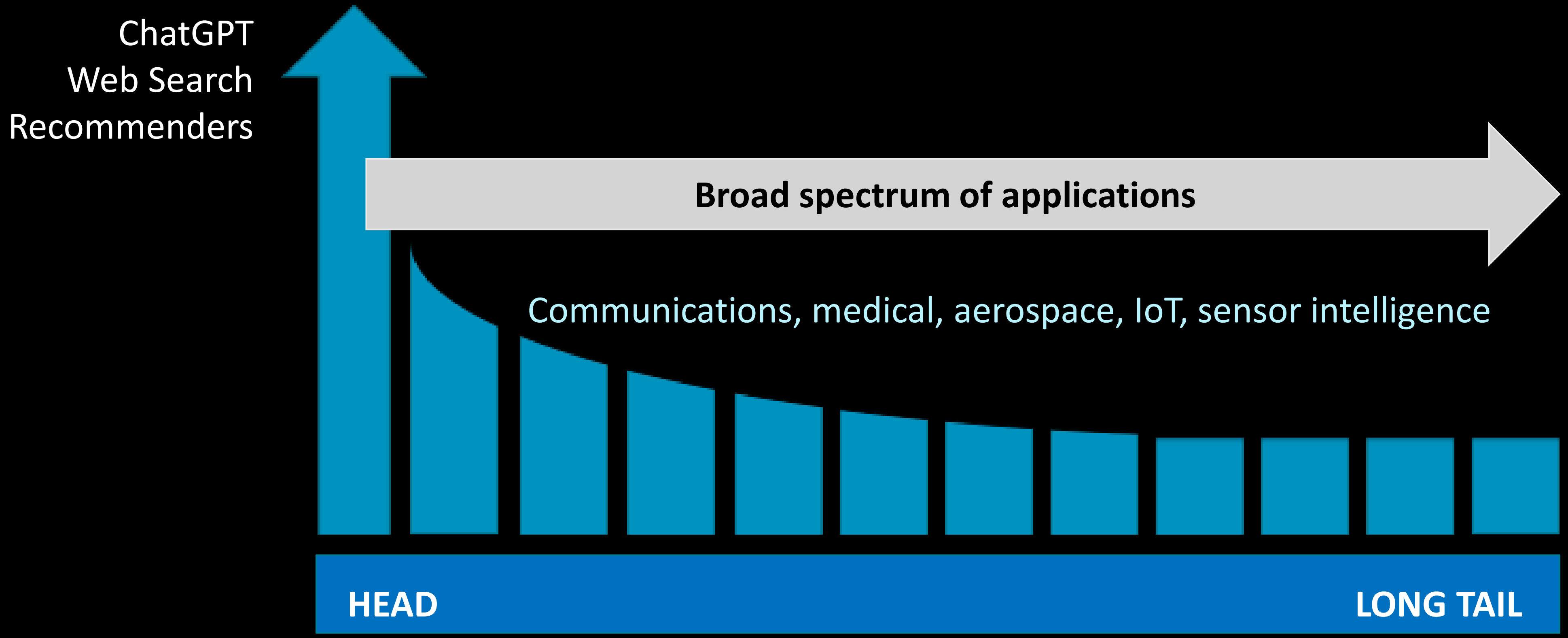
Increasing adoption in many different applications



Tesla AI bot

https://youtu.be/XiQkeWOFwmk?t=6

Pervasive AI



ChatGPT
Web Search
Recommenders

Broad spectrum of applications

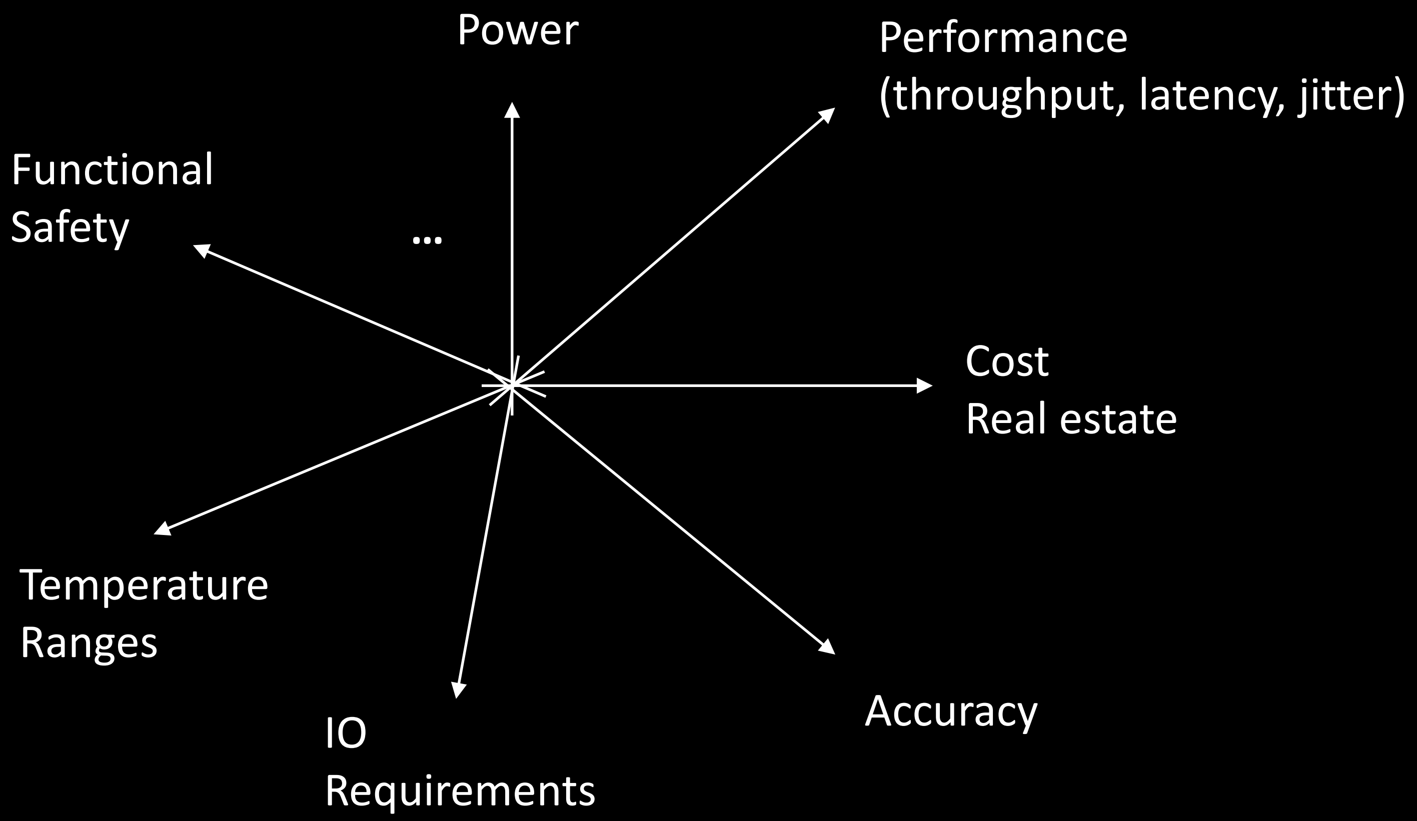
Communications, medical, aerospace, IoT, sensor intelligence

HEAD

LONG TAIL

Adapted from TED Talk: Andrew Ng "How AI could empower any business"

Pervasive AI Comes with Diverse Requirements



Examples of Diverse Requirements

- **IoT/Embedded**
  - Small resource footprint, low power (<10W), low latency (msec), and zero jitter

- **High-Frequency Trading**
  - High-frequency trading (HFT) is an arms race of acquiring data and executing trading decisions fastest
  - Multimillion-dollar advantages through nanosecond differences
  - Extreme low latency requirements (nsec) as DNNs are being adopted for better trading decisions

- **High-Energy Particle Physics**
  - CERN CMS Experiment needs nsec latency for setting recording trigger
  - Incoming data needs to be processed at 7 Tbps
  - Extreme latency requirements (nsec)

Sustainability & Energy Consumption

- Energy footprint on par with whole industrial nations

ChatGPT 4.3 GWh* | Meta AI cluster 53-561 TWh* = Ireland 26 TWh* | Germany 537 TWh*

- Current DNN algorithms represent a **sledgehammer approach**
  - Extremely inefficient

100s kilo Watts matrix multiply

Scope for Improvement: Estimated 10^5

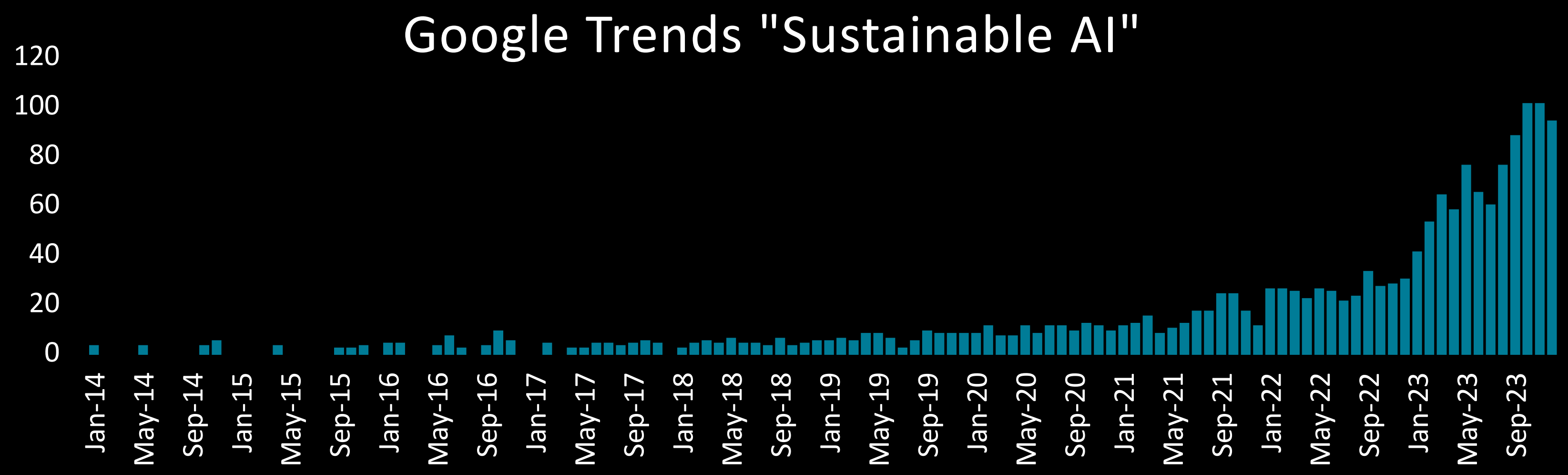
20Watts

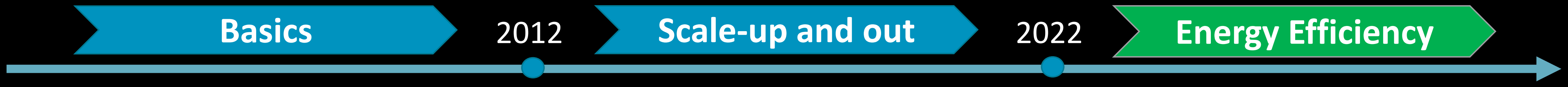
The carbon footprint of ChatGPT. An estimate of the carbon emissions… | by Chris Pointon | Dec, 2022 | Medium
https://www.semianalysis.com/p/meta-discusses-ai-hardware-and-co
Germany - Energy consumption in Germany (worlddata.info)
Ireland - Energy consumption in Ireland (worlddata.info)
**Yu Wang, Tsinghua University, Feb 2016 https://www.numenta.com/blog/2022/05/24/ai-is-harming-our-planet/

*TWh = Tera Watt hours

Paradigm Will Shift towards Energy Efficient AI

- Energy will become the limiting factor for scaling NNs





Basics

2012

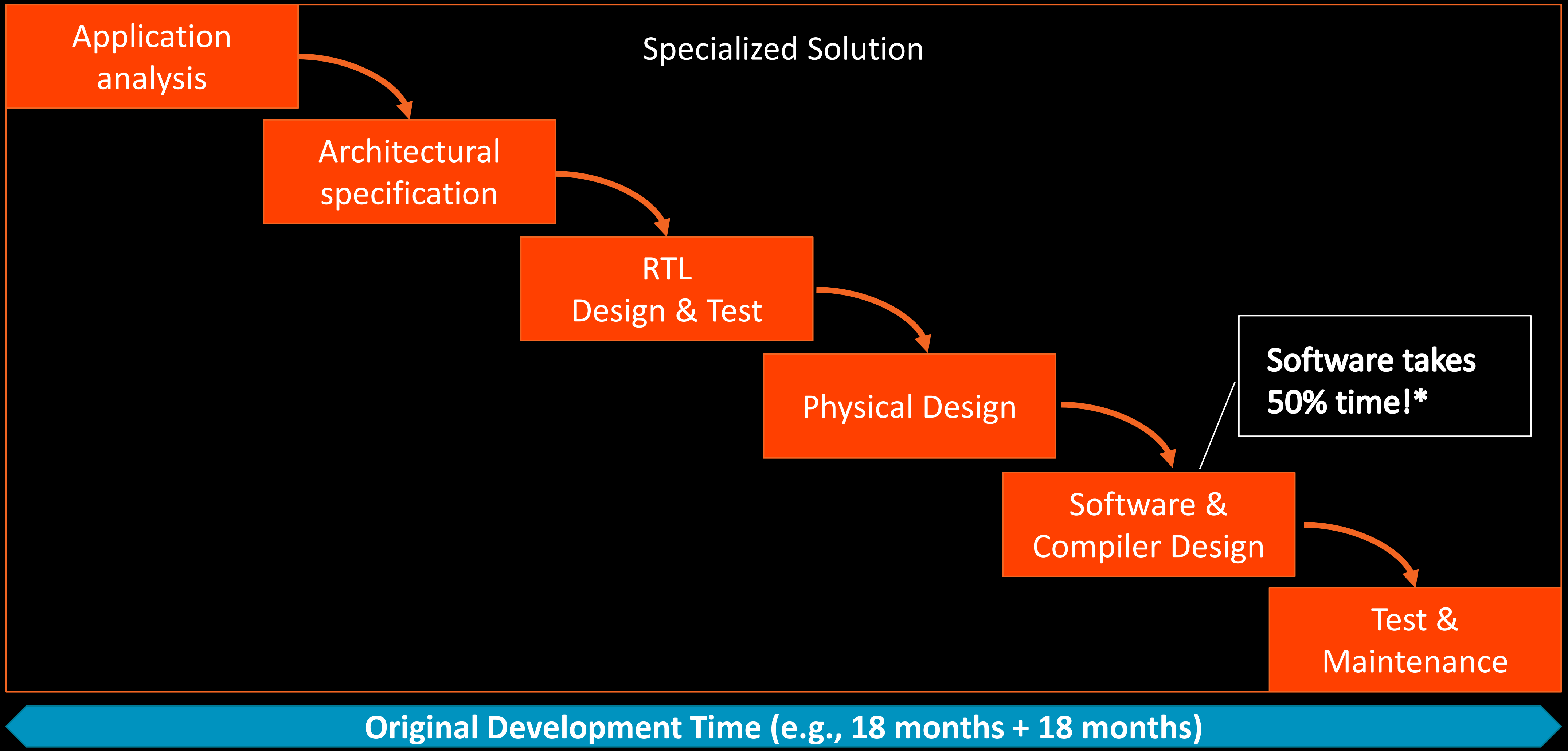
Scale-up and out

2022

Energy Efficiency

Solution Specialization

Classical Hardware Accelerator Design Process (Waterfall)



Specialized Solution

Application analysis

Architectural specification

RTL Design & Test

Physical Design

Software & Compiler Design

Test & Maintenance

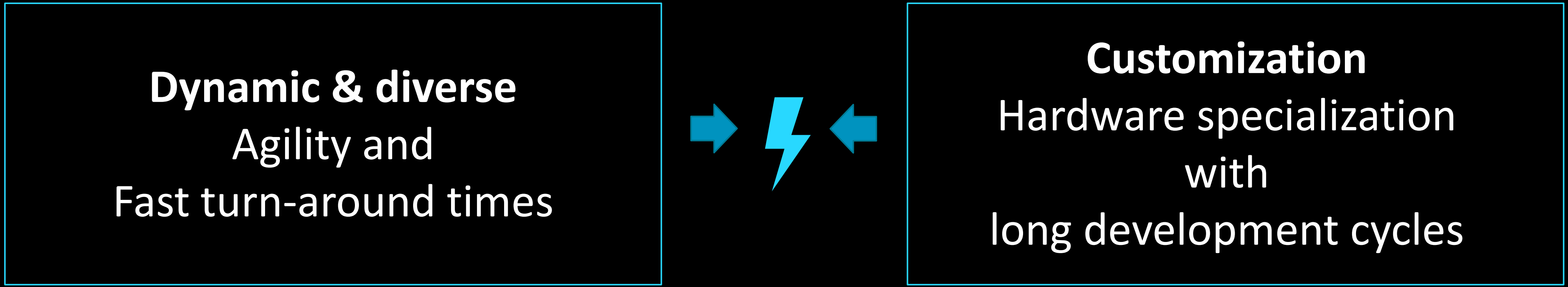
Software takes 50% time!*

Original Development Time (e.g., 18 months + 18 months)

*Source: Chip Design and Manufacturing Cost under Different Process Nodes: Data... | Download Scientific Diagram (researchgate.net)

Challenges in a Nutshell

Dynamic, Diverse & Highly Customized



Dynamic & diverse
Agility and
Fast turn-around times

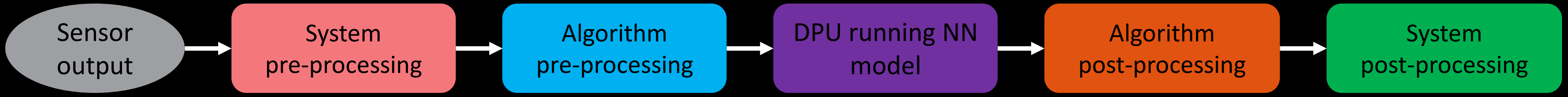
Customization
Hardware specialization
with
long development cycles

Agility in Customization is King

Analyzing Application Requirements

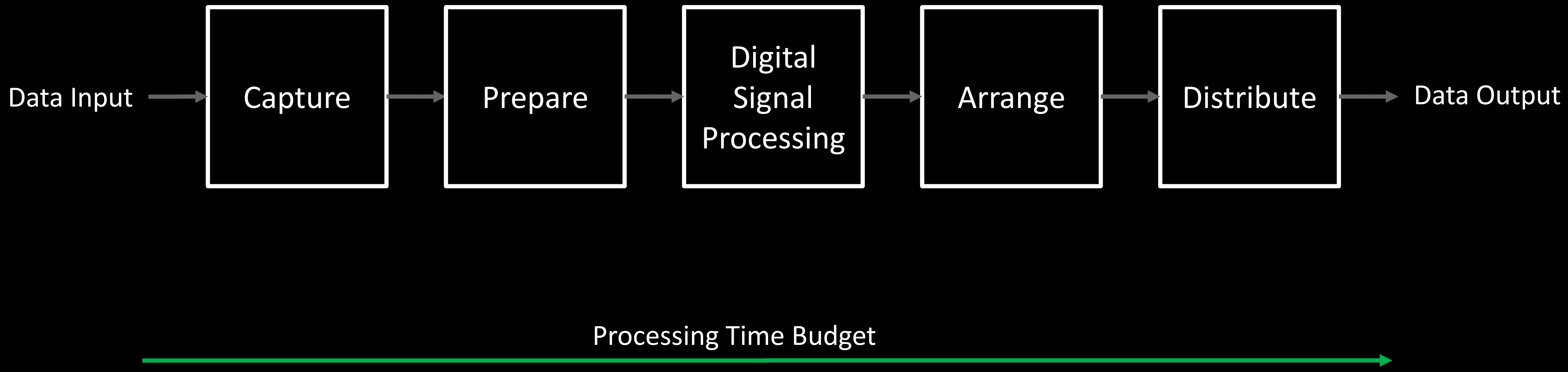
AI Application General Processing Flow

- **A typical abstraction of processing flow:**



- **Algorithm-level processing**
  - Data normalization before sending to DPU
  - Post processing (e.g. bounding boxes decoding in detection)
- **Additional system-level workloads for AI inference**
  - Color conversion / resizing
  - Path planning / control / status update

Typical Signal Processing Scenarios



Decomposing a DSP Algorithm

Key requirements:

- Operators
- Datatype
- Data flow: Balanced/Reduction/Expansion(interpolation etc.)
- Bandwidth (Storage/Pipeline/Distribution)
- Time Budget (Data rate/Processing time/Latency)

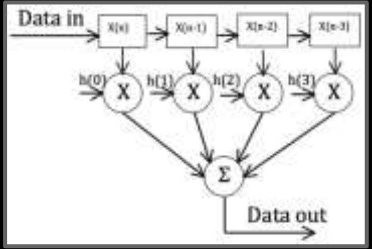
Representative DSP / AI Engine Algorithms

$$\begin{bmatrix} 0 & 2 & 5 & 2 \\ .4 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .6 & 0 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 18 \\ 2 \\ 1 \\ 1.2 \end{bmatrix}$$

Linear Algebra

Matrix-Matrix Multiplication

Matrix-Vector Multiplication



Convolution

FIR Filters

2-D Filters

$$F(x) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi(x\frac{n}{N})}$$

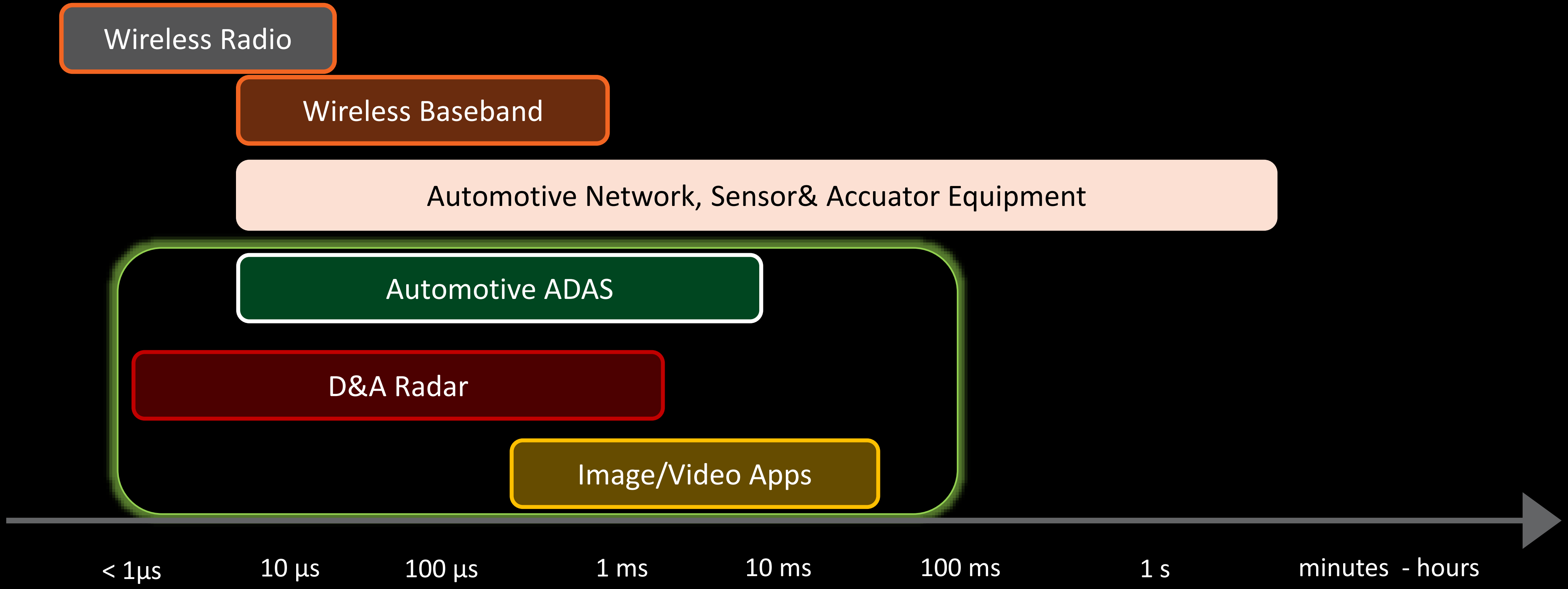
$$f(n) = \frac{1}{N}\sum_{n=0}^{N-1} F(x) e^{j2\pi(x\frac{n}{N})}$$

Transforms

Fast Fourier Transforms (FFTs/iFFTs)

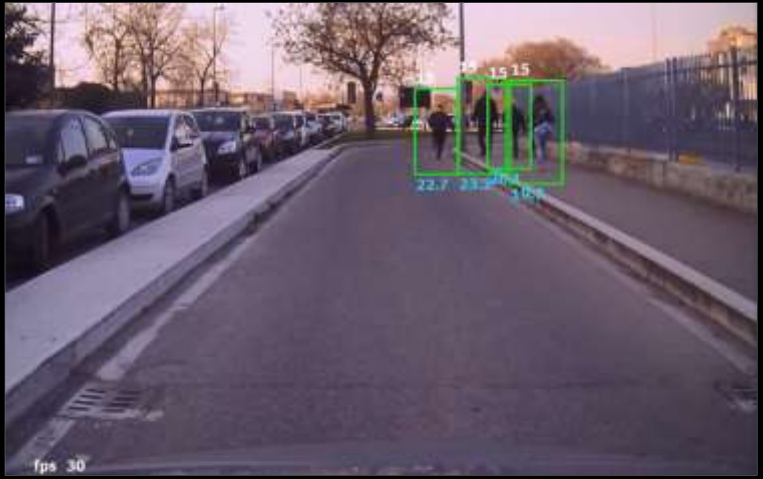
Discrete Cosine Transforms (DCT)

Processing Time Budget

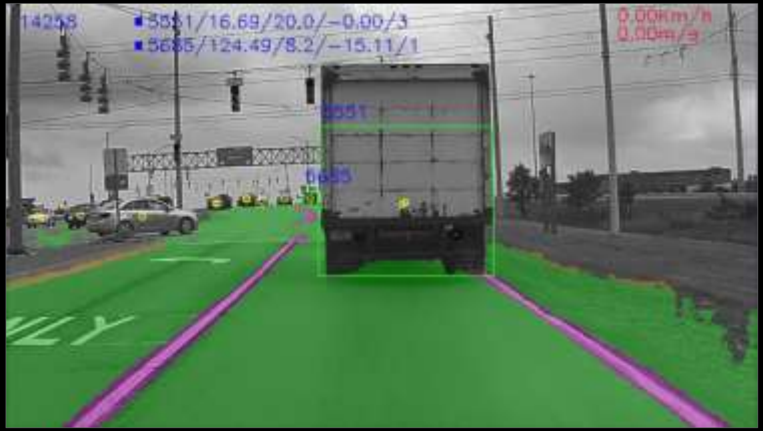


Enabling the Evolution of CV to AI

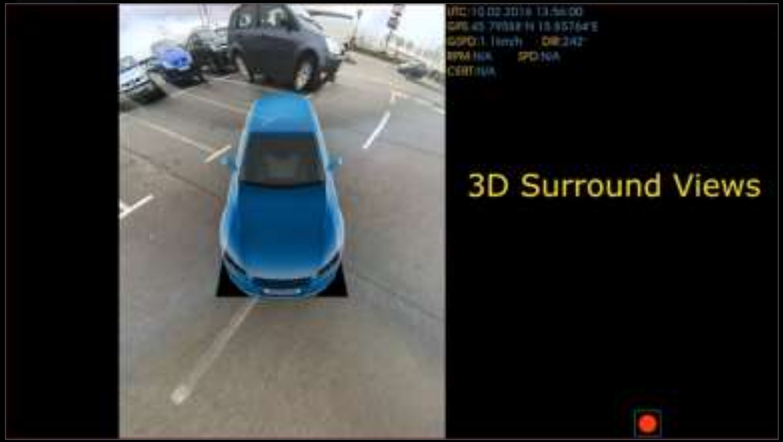
Traditional CV



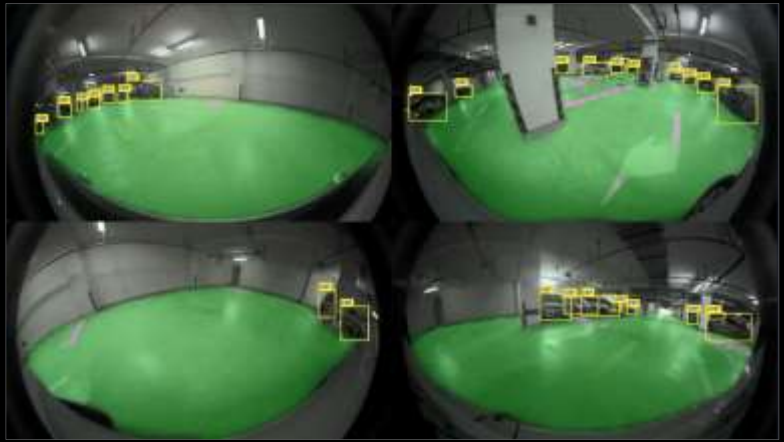
Forward Cam AI



Surround View



Surround View AI



Example ADAS AI Model Support

Classification

- Inception
- Mobilenet
- Resnet
- VGG
- EfficientNet
- MLPerf ResNet50
- OFA ResNet
- Vision Transformer
- Car Type classification
- Car Color classification

Detection

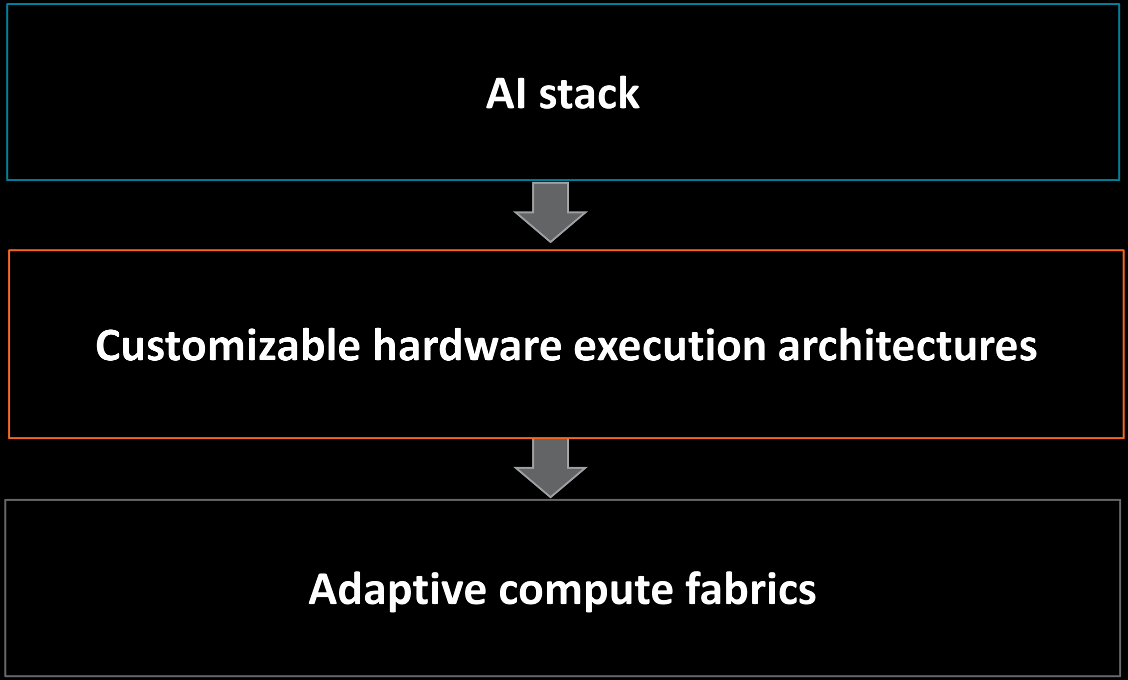
- ssd_mobilenet
- Yolov3
- Yolov4
- YoloX
- Refinedet
- Multi-taskv3
- EfficientDet
- Pointpillars
- Centerpoint
- CLOCs
- Pointpainting
- OFA-Yolo

Segmentation

- ENet
- Semantic FPN
- Salsanext
- Salsanextv2
- SOLO
- HardNet
- Mobilenetv2
- 2D-Unet
- FPN-ResNet18
- Unet-Chaos-CT
- Sa-Gate

Typical Open AI Model support

Enabling Rapid Specialization with Adaptive Compute Fabrics and AI Stacks


AI stack

Customizable hardware execution architectures

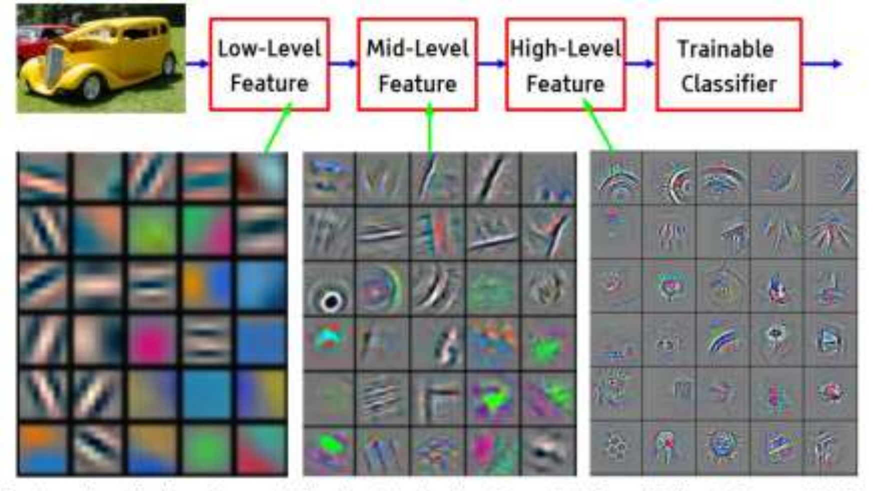
Adaptive compute fabrics

Brevitas

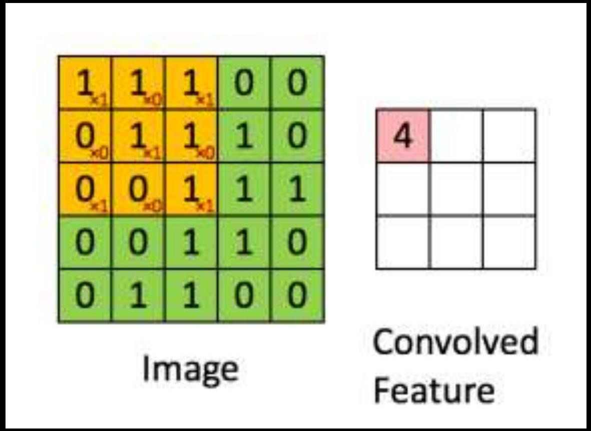
FINN

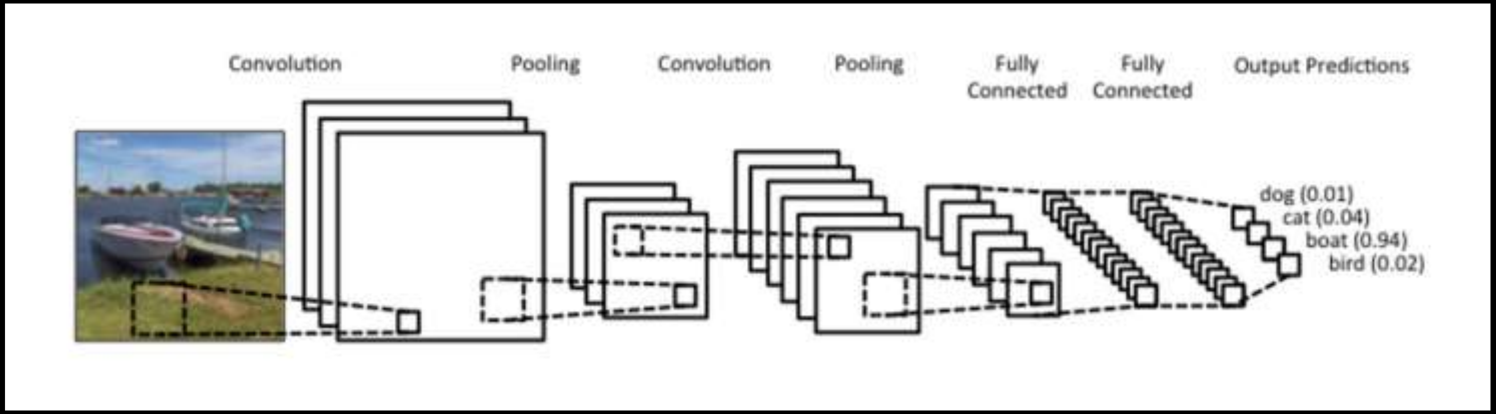
Convolutional Neural Network (CNN)

- A sequence of **convolutional layers** (+ pooling) extracts a **feature map**.
- The final **feature map** is fed to classifier (fully-connected layer) to guess a **class**.

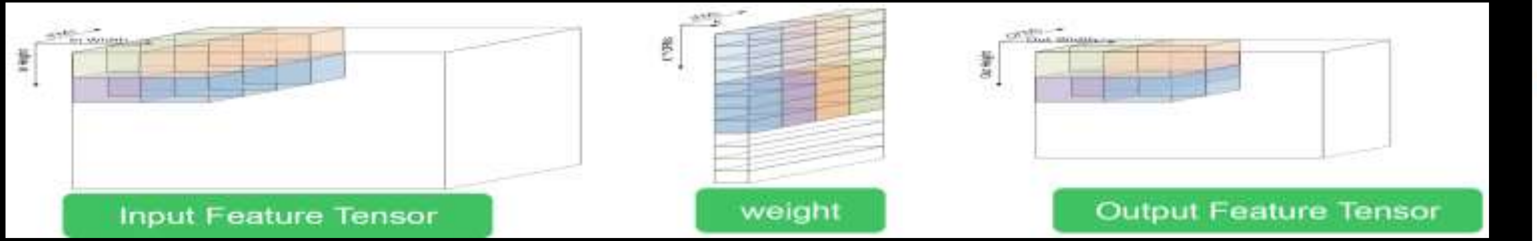


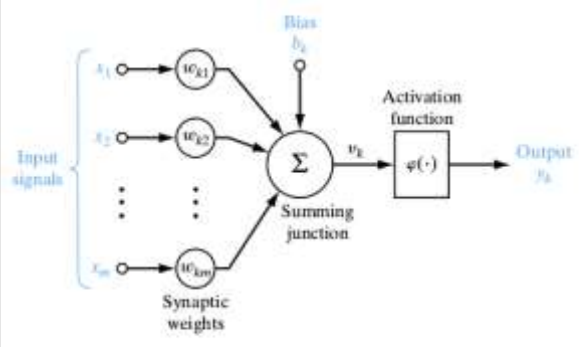
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]



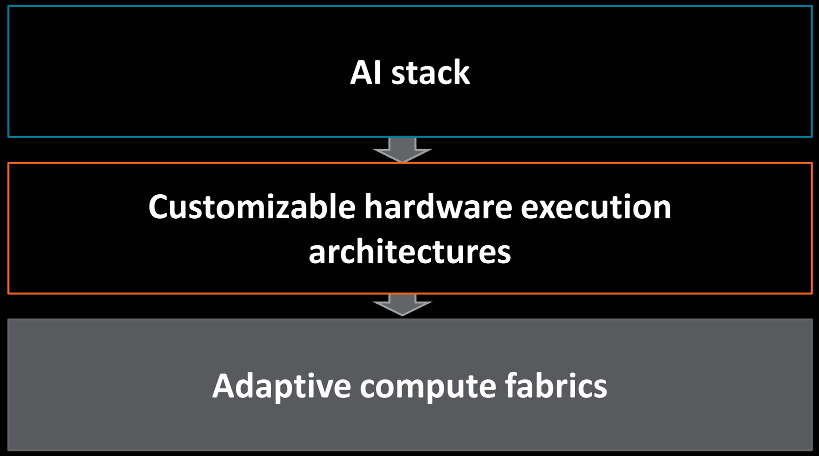


$$\sum_{h}^{H}\sum_{w}^{W} I(i+h, j+w)K(i)(j)$$



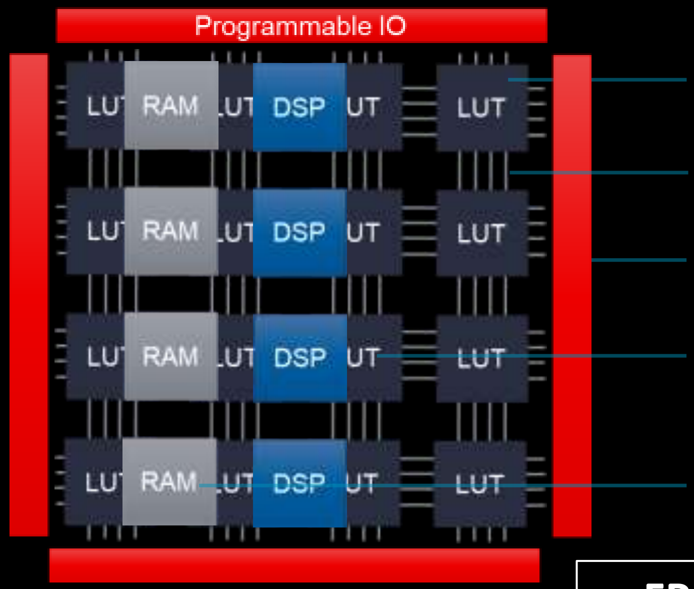


What are adaptive compute fabrics?
FPGAs and AIEs



Primer: Adaptive Computing – FPGAs

- FPGAs are the **chameleon** amongst the semiconductors: flexible, adaptive mostly homogeneous hardware architectures that enable **post-production customization at the architectural level**
- Customize
  - IO interfaces
  - **Functionality post-silicon** (compression, encryption, NN accelerator, key value store,…)
  - **Compute architectures** & **memory subsystems** to meet specific use case’s performance or energy targets



Sea of programmable Lookup Tables (LUTs) ~millions

Programmable Interconnect

Programmable IO

DSPs: n-bit MAC

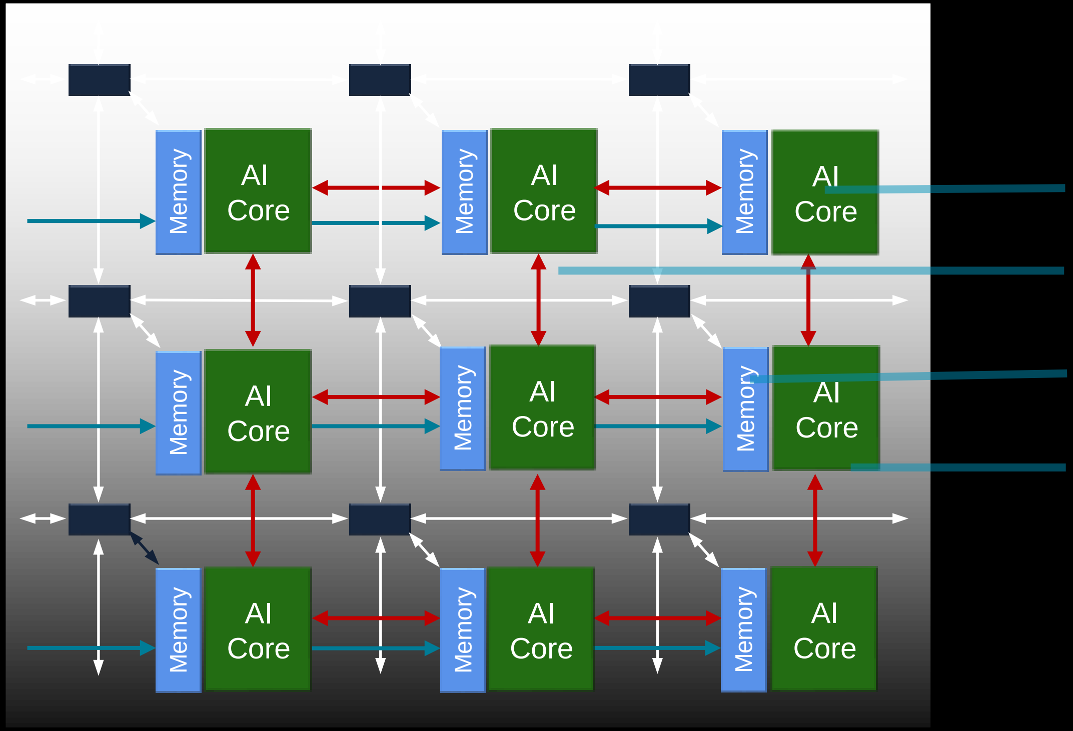
Embedded SRAM ~high bandwidth

FPGAs are flexible and provide the ability to specialize hardware architecture post-production.

Primer: Adaptive Computing – AIEs

- AI Engines (AIEs): new form of higher performant, adaptive compute fabric
  - Higher performance through hardened vector processing in VLIW cores, just word-based (instead of bit-based) with native support for ML-optimized data types (e.g., INT8, block float,…)
  - Great flexibility because of interconnectivity and separate control flow

    => **adapt the execution architecture to different workloads**



Matrix of VLIW/SIMD vector processors (10s...100x)

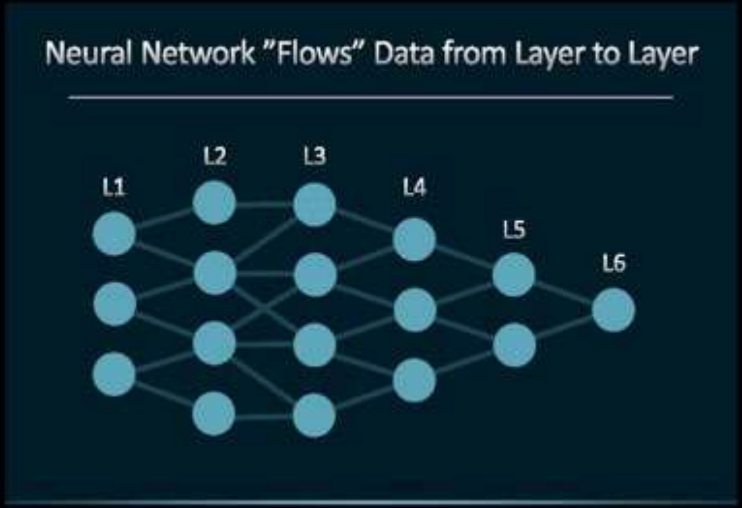
Flexible interconnect

Tightly coupled, embedded memory (1..10s MB)

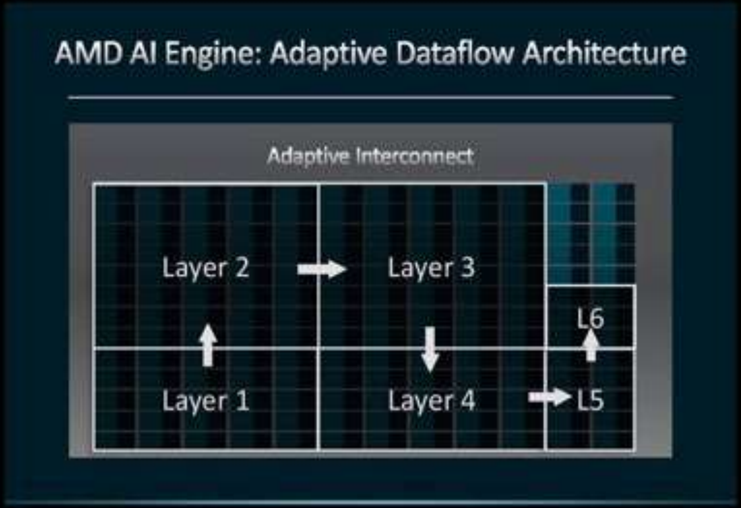
AIE are software compiled and don't require synthesis

AI Model mapping into AIE

High Performance, Energy Efficient, Customizable for AI Workloads

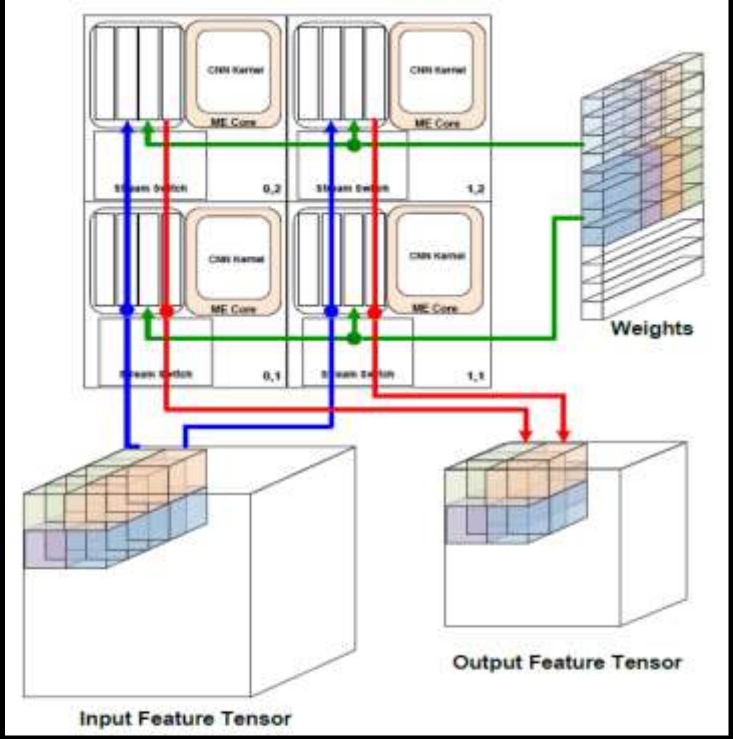


Neural Network "Flows" Data from Layer to Layer

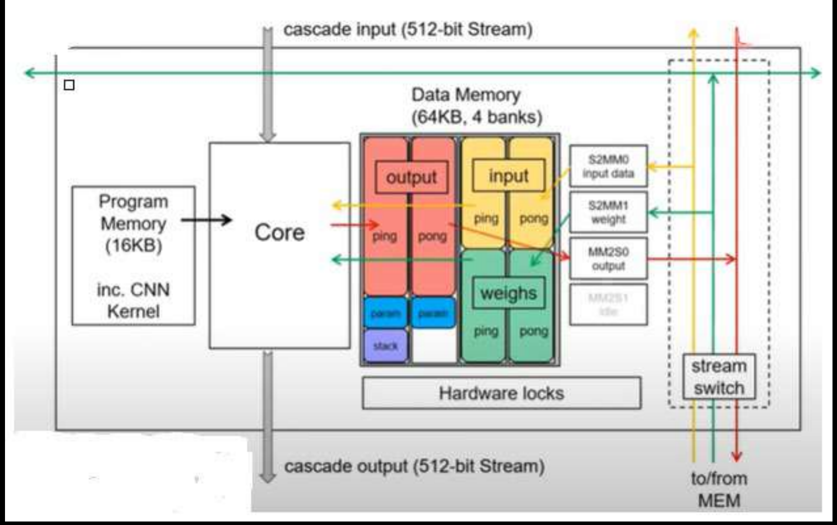


AMD AI Engine: Adaptive Dataflow Architecture

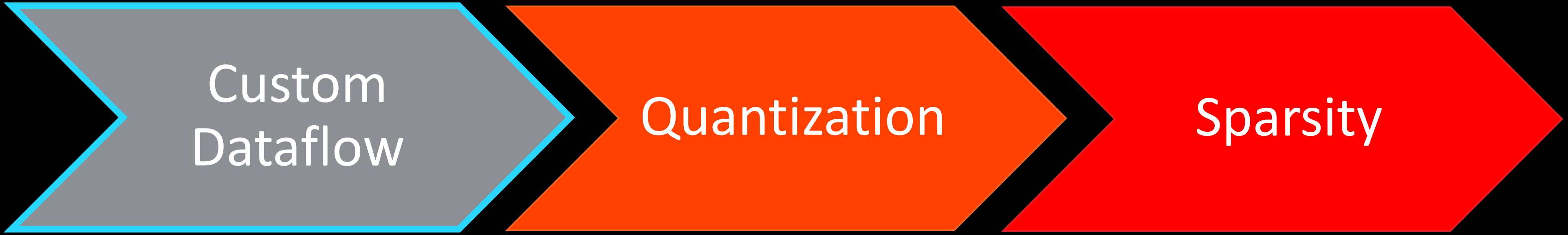
Tensor Broadcasting in AIE Array:



* Use Case: Mapping CNN to AIE Tile:



Key Concepts



Customized for specific topologies

Customized in data types

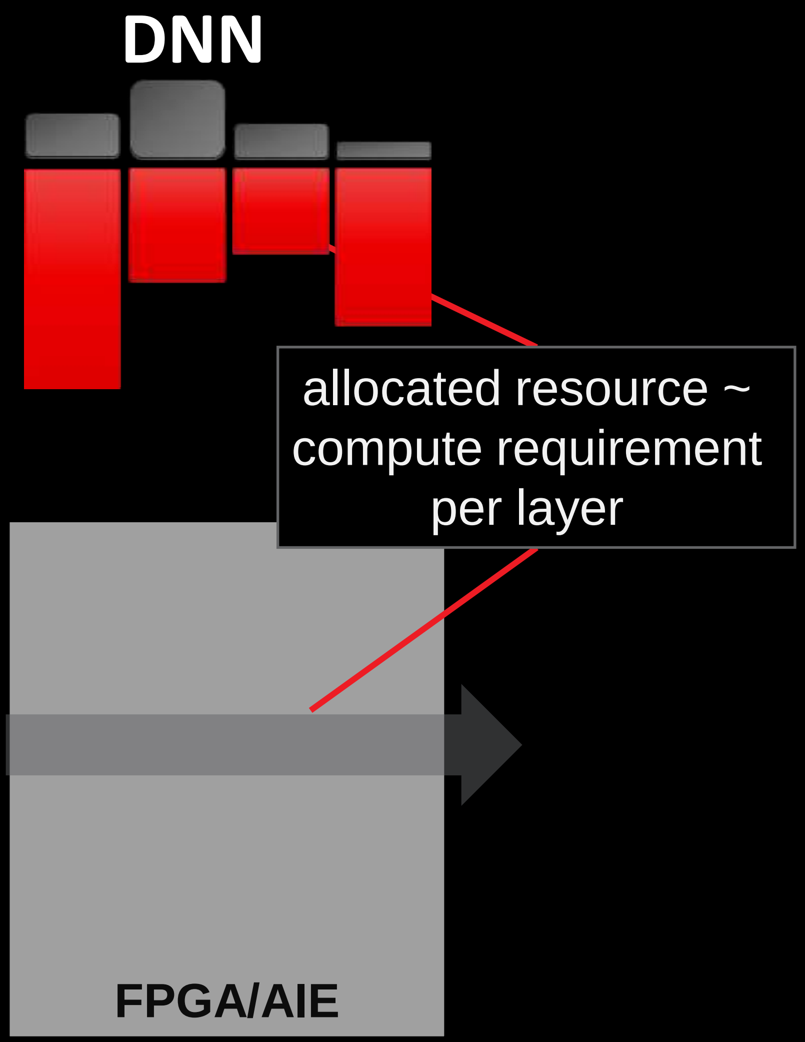
Sparse neural circuits

Dataflow - Specializing for Individual Topologies

- Hardware instantiates the topology as a dataflow architecture

- Customize everything to the specifics of the given DNN, its operations and connectivity

- Benefits: energy efficiency, latency and throughput scalability



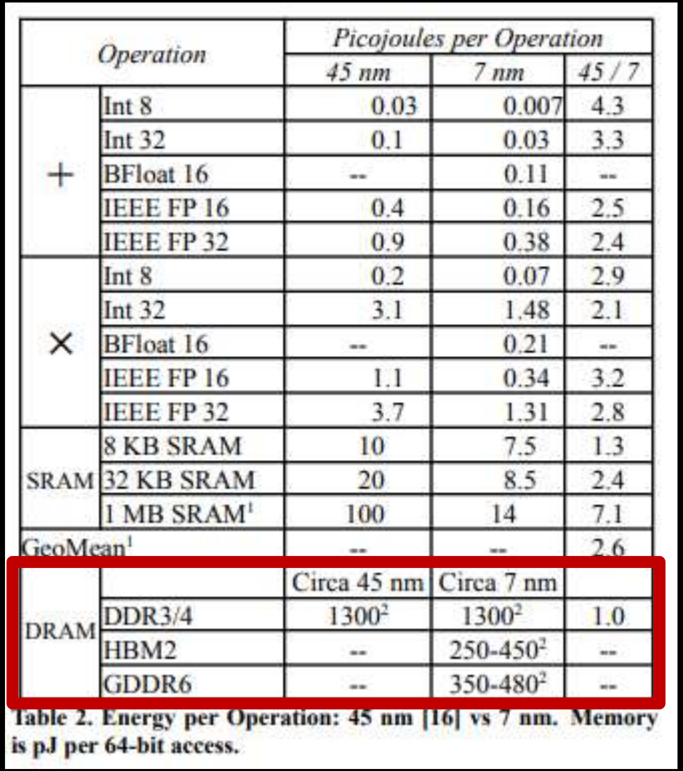
allocated resource ~ compute requirement per layer

Dataflow - Energy Efficiency

- Architecture only computes and stores what’s needed in the specific use case
  - Customized memory and compute subsystem

- Minimizes movement & storing of data
  - Activations are not buffered externally; they are in SRAM and registers moved directly from one layer to next

- High efficiency through concurrent communication and compute
  - Each layer starts computing as soon as first inputs are available
  - Shortens execution time => energy saving ($E = P * time$)

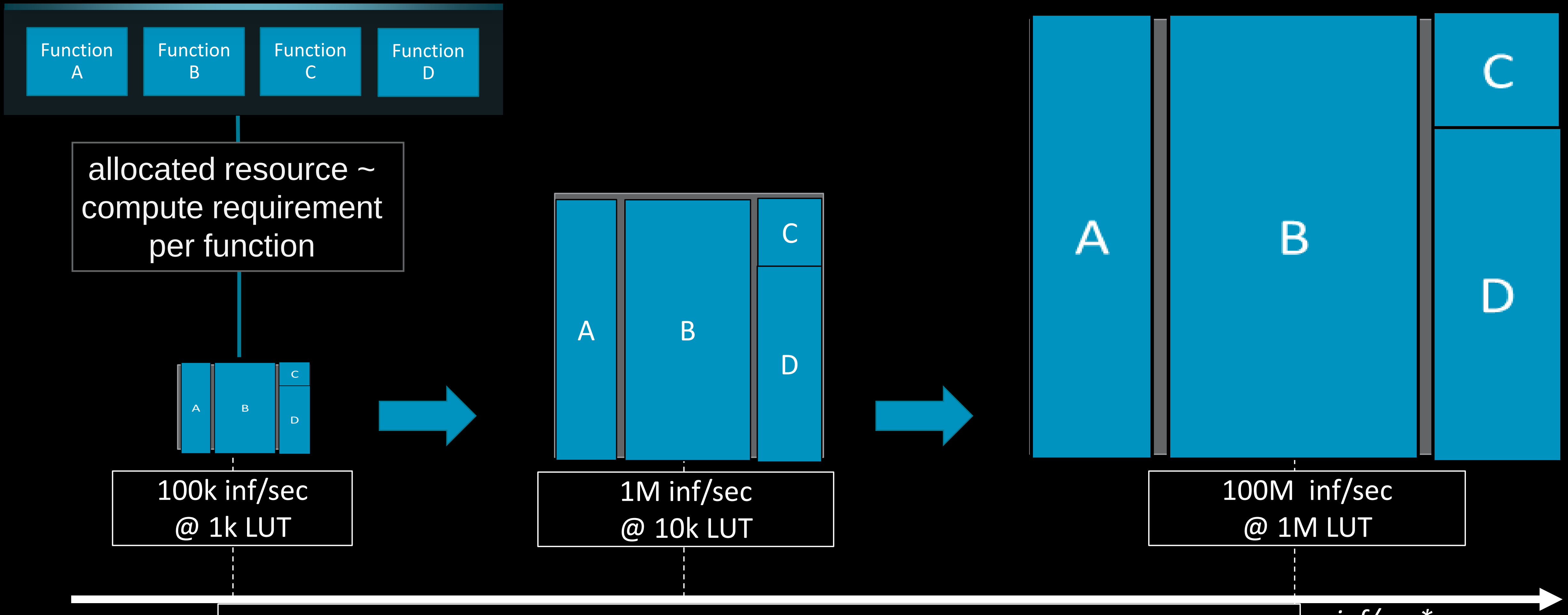


Operation		*Picojoules per Operation*		
		45 nm	*7 nm*	*45 / 7*
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM[1]	100	14	7.1
GeoMean[1]		--	--	2.6
DRAM		Circa 45 nm	Circa 7 nm	
	DDR3/4	1300[2]	1300[2]	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

Table 2. Energy per Operation: 45 nm [16] vs 7 nm. Memory is pJ per 64-bit access.

Jouppi, Norman P., et al. "Ten lessons from three generations shaped Google’s TPUv4i: *ISCA’*2021.

Dataflow - Adapt and Scale to Diverse Workloads



allocated resource ~ compute requirement per function

100k inf/sec
@ 1k LUT

1M inf/sec
@ 10k LUT

100M inf/sec
@ 1M LUT

Dataflow can scale performance & resources to meet diverse requirements
Without batching!

*Inf/sec: inferences/second

Key Concepts

Custom Dataflow

Quantization

Sparsity

Customized for specific Topologies

Customized in data types

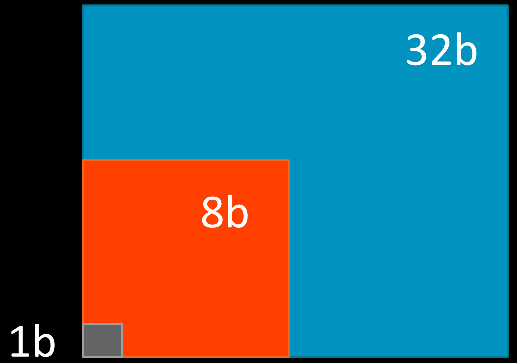
Sparse neural circuits

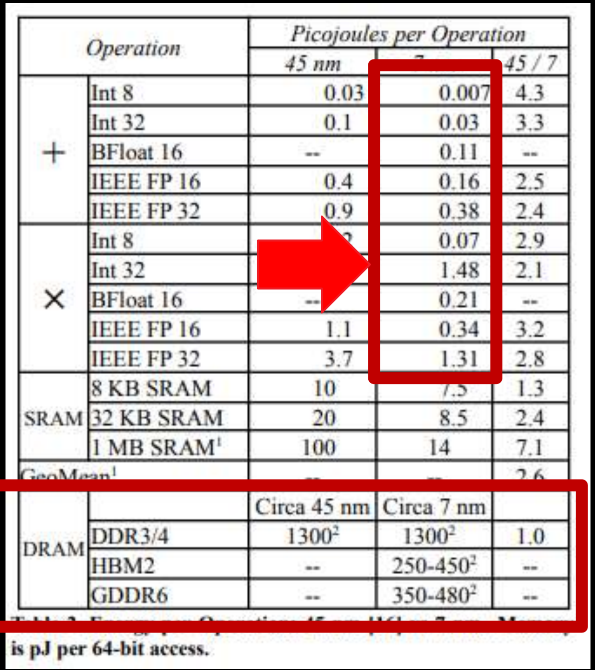
Customizing Arithmetic to Minimum Precision

Quantization

- Reducing precision shrinks hardware cost/scales performance
  - For integer datatypes, LUT cost proportional to bitwidths in weight and activations (e.g., INT1 : INT8: 70x)
  - Instantiate n-times more compute within the same fabric, thereby scale performance n-times or shrinks hardware cost

- Energy
  - Faster execution => less energy ($E = P * time$)
  - Using reduced precision operators saves energy
  - Reduces memory footprint
    - ResNet50 @ 32b: 102.5 MB, ResNet50 @ 2: 6.4 MB
    - NN model can stay on-chip => no external memory access => saves energy





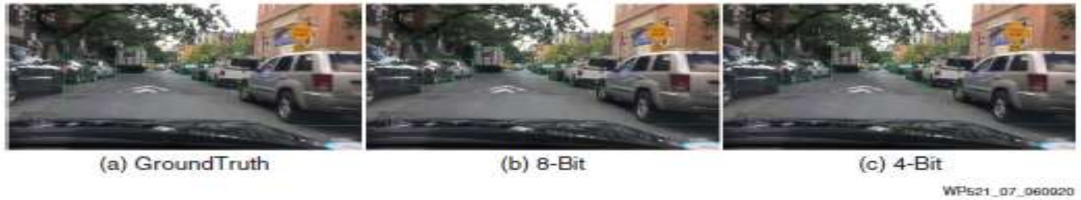
Operation		Picojoules per Operation		
		45 nm	7 nm	45 / 7
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	[illegible]	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM[1]	100	14	7.1
GeoMean[1]		--	--	2.6
DRAM		Circa 45 nm	Circa 7 nm	
	DDR3/4	1300[2]	1300[2]	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

Table 2. Energy per Operation: 45 nm [16] vs 7 nm. Memory is pJ per 64-bit access.

Jouppi, Norman P., et al. "Ten lessons from three generations shaped google's tpuv4i: *ISCA'*2021.

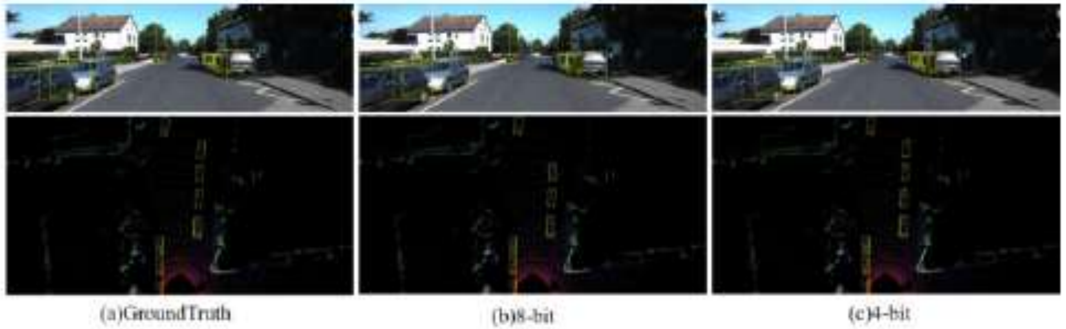
Low precision perception

(Convolutional Neural Network with INT4 Optimization on Xilinx Devices, WP521 (v1.0.1) June 24, 2020)



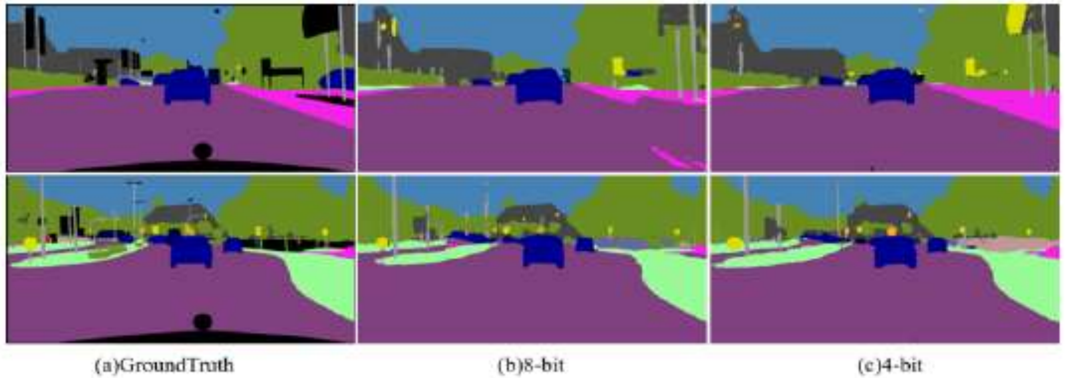
(a) GroundTruth (b) 8-Bit (c) 4-Bit

Figure 7: **Visualization of 2D Detection**



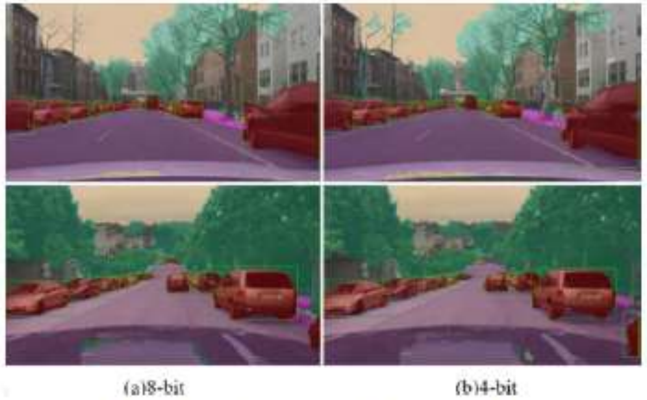
(a)GroundTruth (b)8-bit (c)4-bit

Figure 8: **Visualization of 3D Detection on Camera and Bird-Eye-View**



(a)GroundTruth (b)8-bit (c)4-bit

Figure 9: **Visualization of Semantic Segmentation**



(a)8-bit (b)4-bit

Figure 10: **Visualization of Multi-Task Learning**

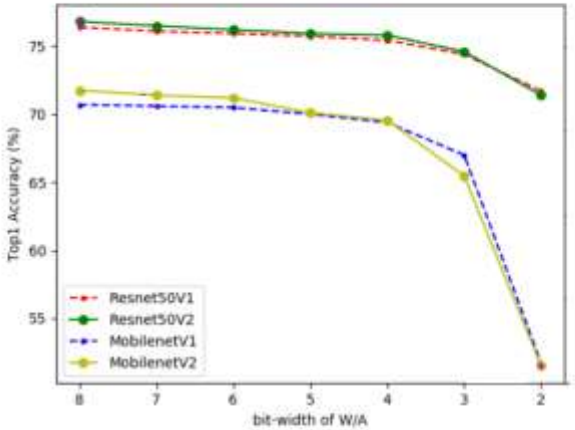
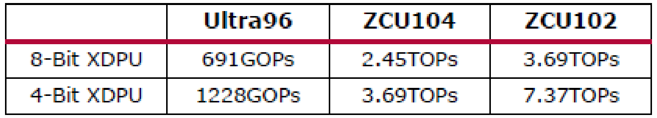
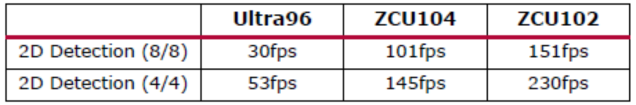


Table 7: **Performance Comparison between 4-Bit XDPU and 8-Bit XDPU**



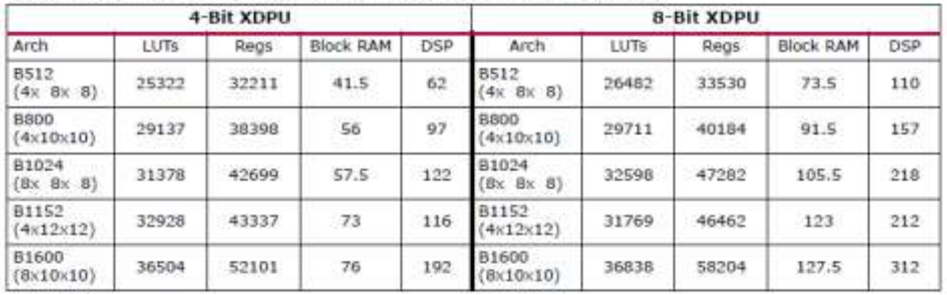
	Ultra96	ZCU104	ZCU102
8-Bit XDPU	691GOPs	2.45TOPs	3.69TOPs
4-Bit XDPU	1228GOPs	3.69TOPs	7.37TOPs

Table 9: **Frame Rate between 4-Bit DPU and 8-Bit DPU**



	Ultra96	ZCU104	ZCU102
2D Detection (8/8)	30fps	101fps	151fps
2D Detection (4/4)	53fps	145fps	230fps

Table 8: **Resource Comparison between 4-Bit XDPU and 8-Bit XDPU**



4-Bit XDPU					8-Bit XDPU				
Arch	LUTs	Regs	Block RAM	DSP	Arch	LUTs	Regs	Block RAM	DSP
B512 (4x 8x 8)	25322	32211	41.5	62	B512 (4x 8x 8)	26482	33530	73.5	110
B800 (4x10x10)	29137	38398	56	97	B800 (4x10x10)	29711	40184	91.5	157
B1024 (8x 8x 8)	31378	42699	57.5	122	B1024 (8x 8x 8)	32598	47282	105.5	218
B1152 (4x12x12)	32928	43337	73	116	B1152 (4x12x12)	31769	46462	123	212
B1600 (8x10x10)	36504	52101	76	192	B1600 (8x10x10)	36838	58204	127.5	312

Key Concepts

Custom Dataflow

Quantization

Sparsity

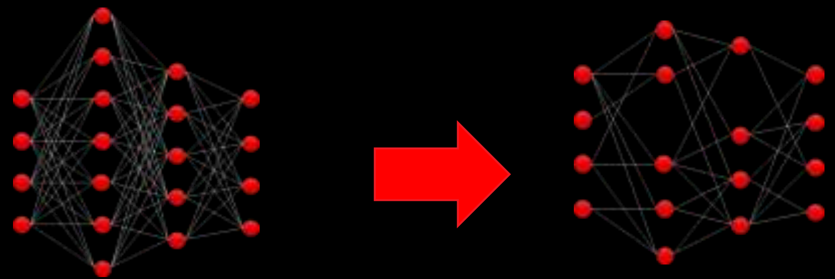
Customized for specific Topologies

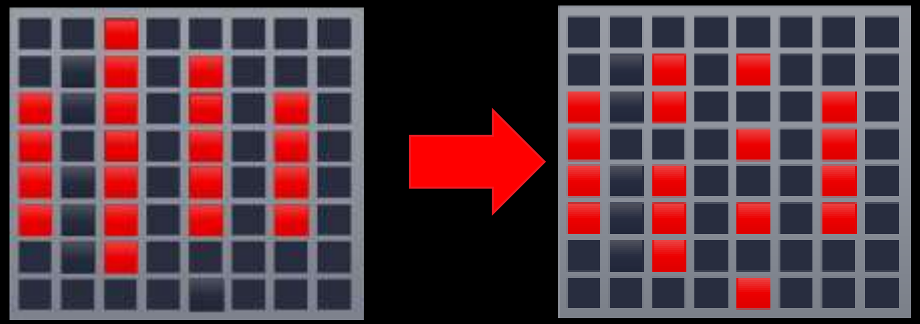
Customized in data types

Sparse neural circuits

Sparsity – Energy Efficiency

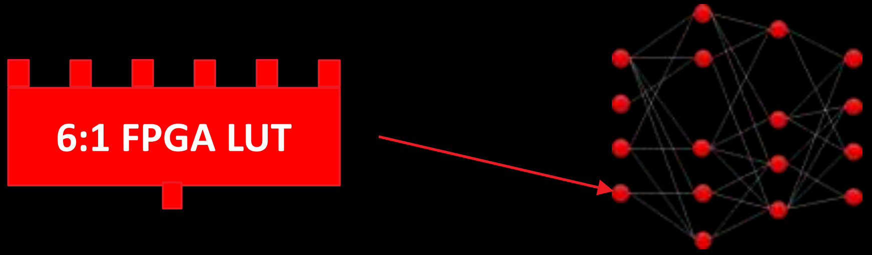
- DNNs are naturally sparse
- Massive scope to improve ML efficiency through sparsity
  - The human brain is highly sparse (98%) & operates on the power of a light bulb (~20W)*
- Sparse topologies result in irregular compute patterns which are difficult to accelerate on vector- or matrix-based execution units
  - Poor efficiency
- With streaming dataflow architectures, where every neuron and synapse is represented in the hardware, we can maximize efficiency



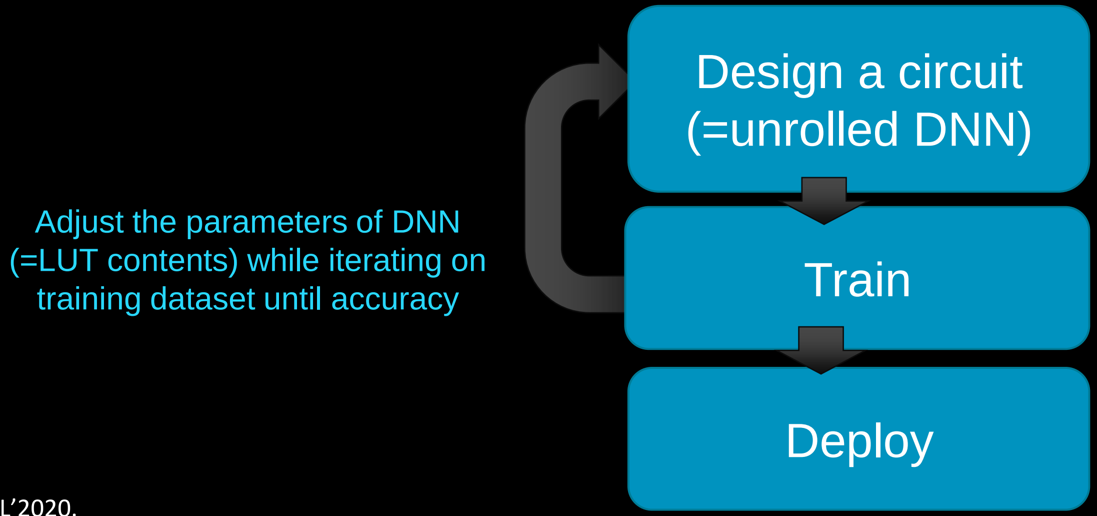


Sparsity – Extreme Codesign with FINN LUT mapping -> LogicNets

- **Idea**
  - A LUT in an FPGA can represent a neuron
  - Design a highly sparse circuit in an FPGA
  - Represent this as a DNN to the training framework
  - Learn the LUT contents



High-efficiency and maximum performance by design (classification at clock rate)


Design a circuit
(=unrolled DNN)
Train
Deploy
Adjust the parameters of DNN (=LUT contents) while iterating on training dataset until accuracy

*https://www.numenta.com/blog/2022/05/24/ai-is-harming-our-planet/

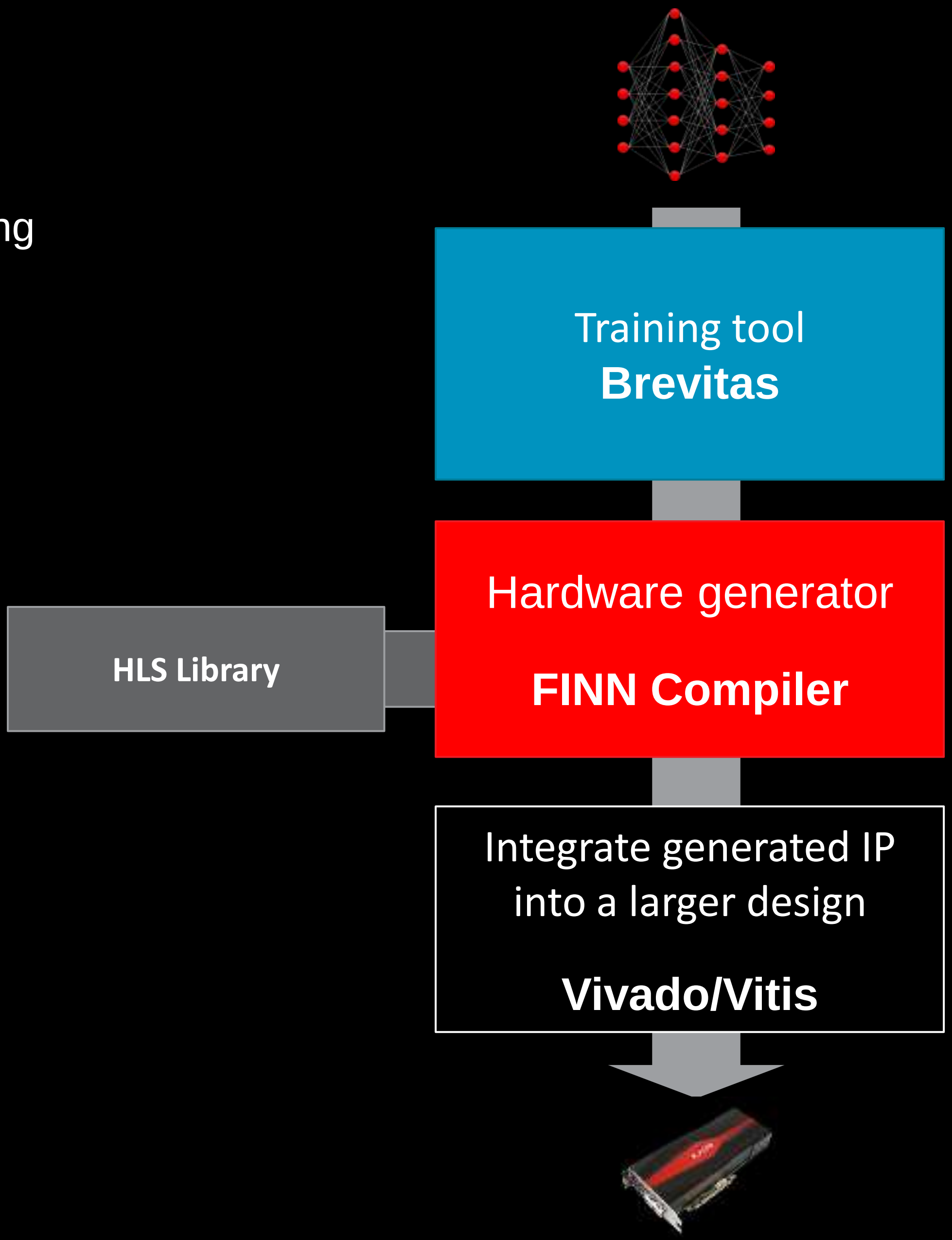
Umuroglu, Yaman, et al. "LogicNets: co-designed neural networks and circuits for extreme-throughput applications." FPL’2020.

Example: FINN & Brevitas

- End-to-end flow – from DNN to bitstream
  - Enables generation of highly customized hardware architectures using **quantization** and **dataflow** and **fine-granular sparsity**

- Components
  - Training tool: Brevitas
  - Hardware generator (FINN)
    - Kernel library (HLS)

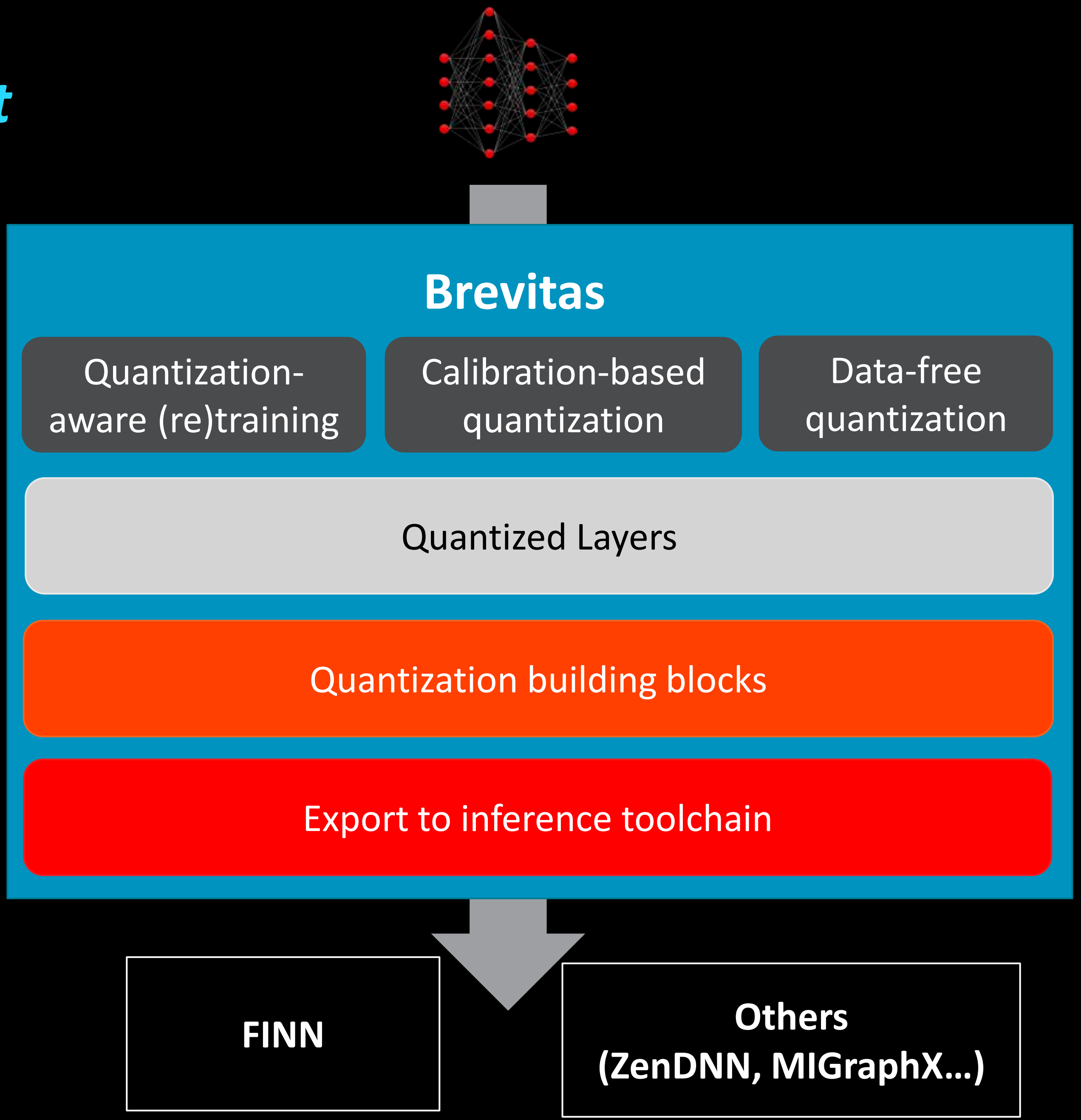
- Open-source
  - Easy collaboration with customers
  - Flexibility to adapt to fast-moving application space
  - Third-party contributions



Brevitas - PyTorch Library

Offering Agile Quantization Support

- First class support for custom data types and operators at ML framework level
  - Arbitrary precision integer, float, block-style quantization
  - Extendible to user-defined datatypes and operators and support for any hardware-specific datatype at training
- Composable building blocks at multiple abstraction levels that can be arbitrarily combined
- Integration with different compiler stacks
  - Exports commonly used representation format (for example ONNX)



Brevitas

Quantization-aware (re)training

Calibration-based quantization

Data-free quantization

Quantized Layers

Quantization building blocks

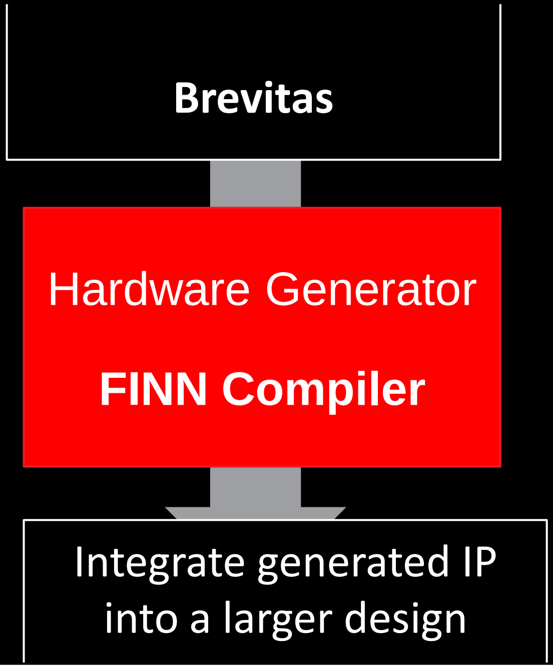
Export to inference toolchain

FINN

Others (ZenDNN, MIGraphX…)

FINN Compiler

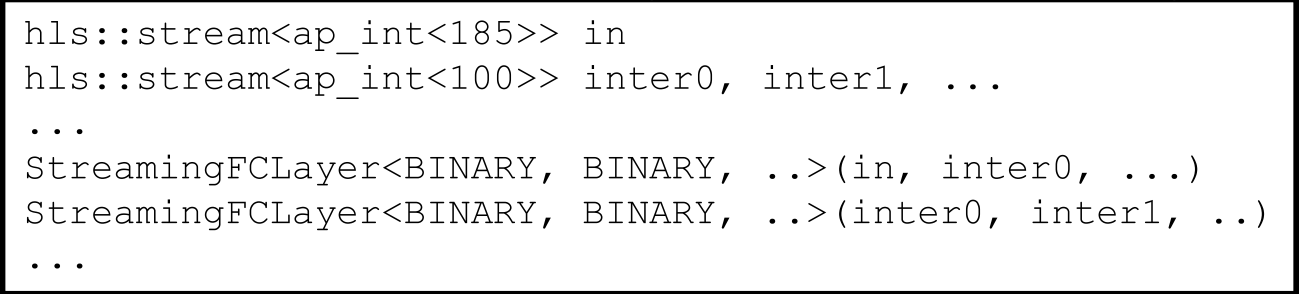
- Modular **graph compiler** with well-defined abstraction levels
- Incrementally lowers ONNX graph to a hardware description through **transformations**
- Performs **optimizations**
  - Layer fusion
- Explores the **design space**
  - Calculates the degrees of parallelism for each kernel using resource cost and performance models
- **Code-generate**s a dataflow C++ description using the parameterizable **kernel library**
- Creates **DNN hardware IP**

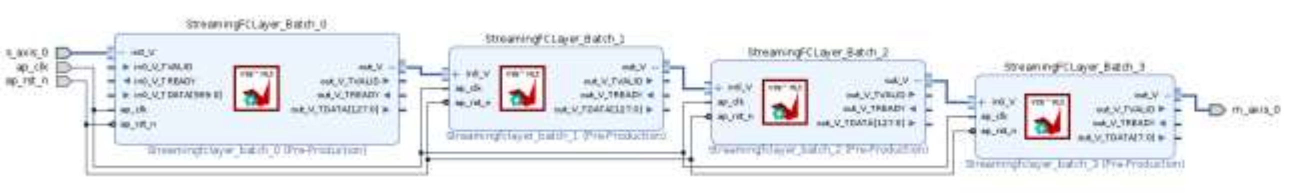

Brevitas

Hardware Generator

FINN Compiler

Integrate generated IP into a larger design

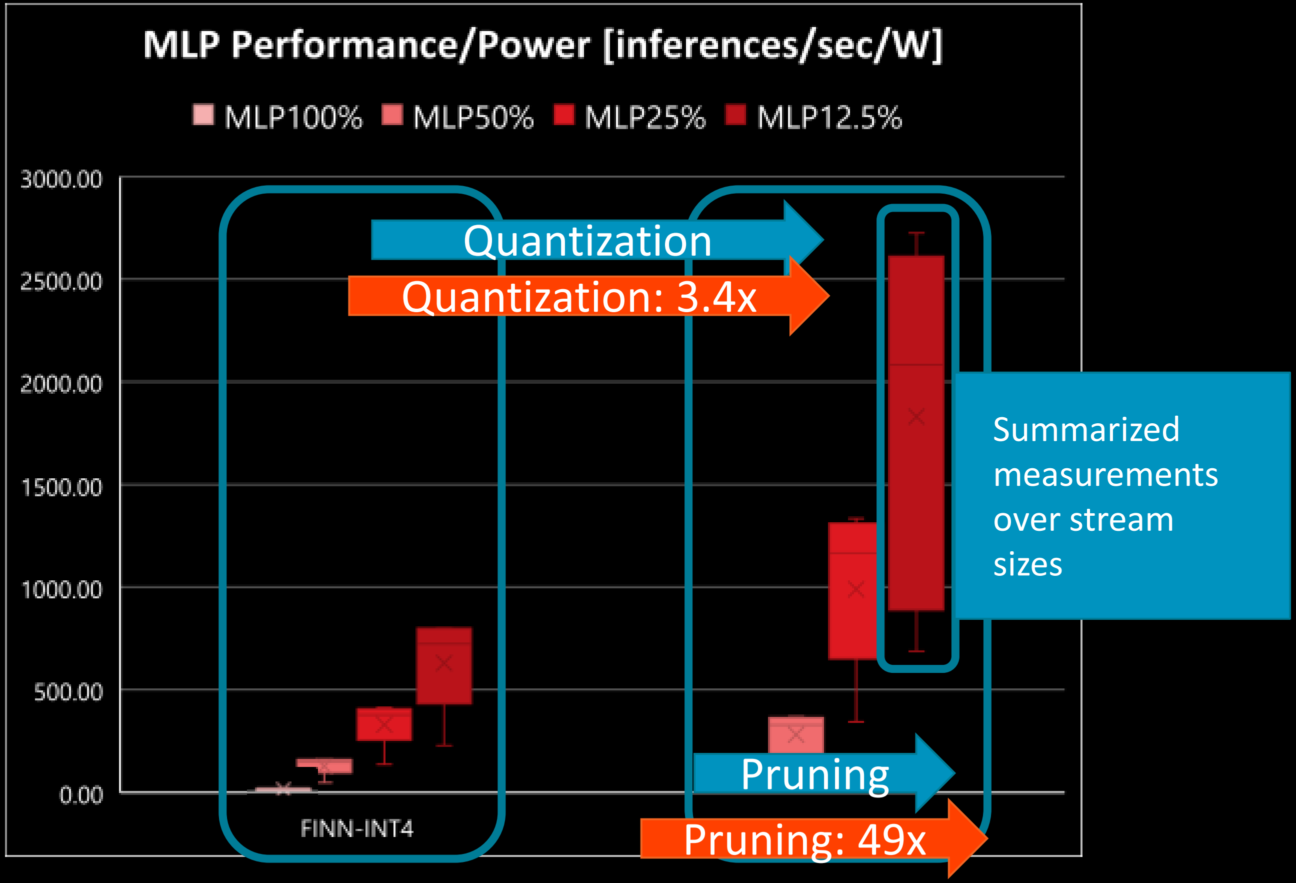

```
hls::stream<ap_int<185>> in
hls::stream<ap_int<100>> inter0, inter1, ...
...
StreamingFCLayer<BINARY, BINARY, ..>(in, inter0, ...)
StreamingFCLayer<BINARY, BINARY, ..>(inter0, inter1, ..)
...
```



Some Example Results

Energy Efficiency through Quantization and Sparsity

- Benchmarking activity* across topologies, devices, and optimization schemes
- Example representing typical behavior: one MLP and one CNV, using quantization & pruning on an FPGA (FINN)



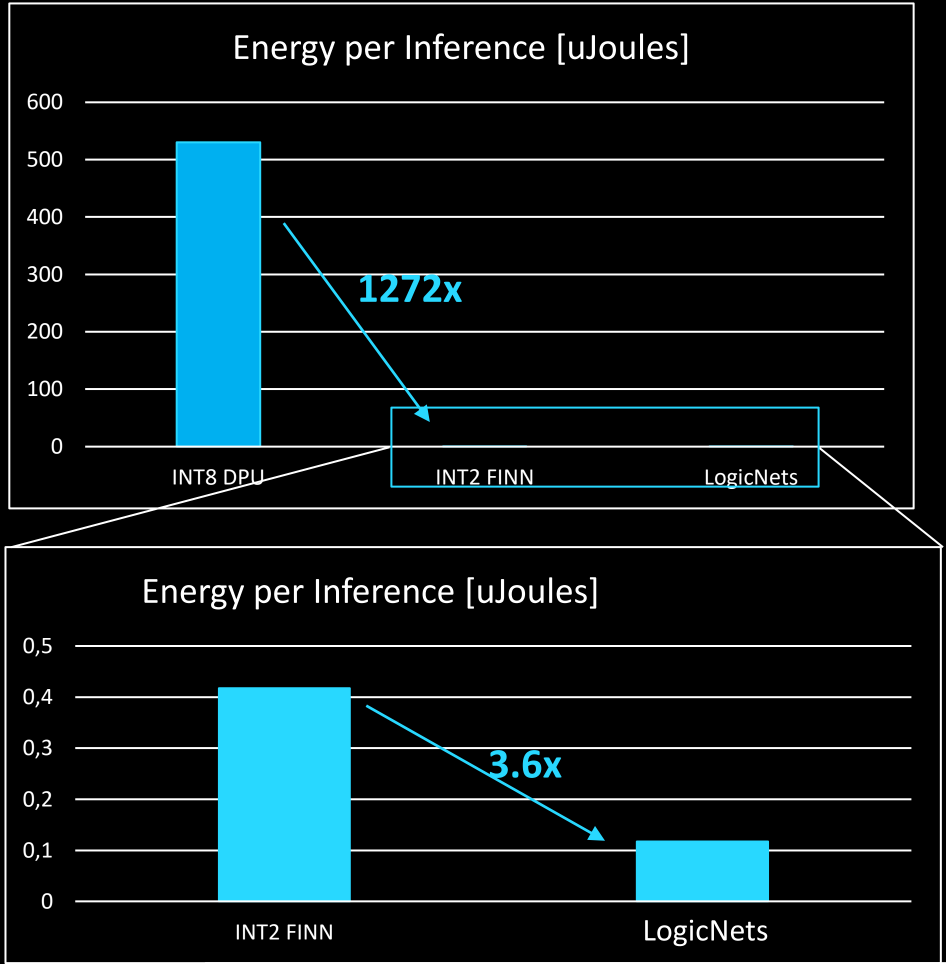
Summarized measurements over stream sizes

Significant energy efficiency through pruning and quantization on FPGAs possible

*QuTiBench (rcl-lab.github.io)
http://www.tara.tcd.ie/handle/2262/96391

Energy Efficiency: FINN & LogicNets (One bit LUT - FINN)

Results Demonstrate the Potential



Reducing precision & Dataflow => 1272 improvement

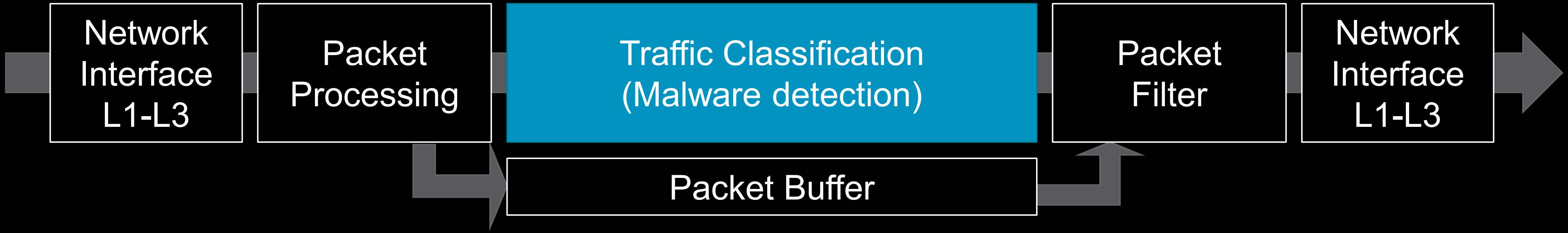
LogicNets: 3.6x over FINN

Details:
Network Security Application
Malware Classifier
UNSW dataset
MLP 92k Ops/inference
INT8 with VitisAI,
INT2 with Brevitas and FINN
Board power ZCU104

Energy calculated
LogicNets assume

Total: ~4500x Energy Improvement through Post-Silicon Hardware Specialization
Much more work coming...

Cyber Security – Line-rate Classification with Nanosecond Latency



- FINN implementation of UNSW-NB15 malware classifier
  - 2b weights & activations
  - 91.9% accuracy
  - 300M inferences/sec with 18 nsec latency
  - 8k LUT

- FINN implementation of DDoS classifier trained on CIC-IDS2017 dataset
  - 2b weights & activations
  - 85% F1-score (binary classification using flow-based per-packet features)
  - 19.2M inferences/sec, 52nsec latency
  - 18.6K LUTs

Work in progress:
Expected to scale to 300M inferences/sec too …

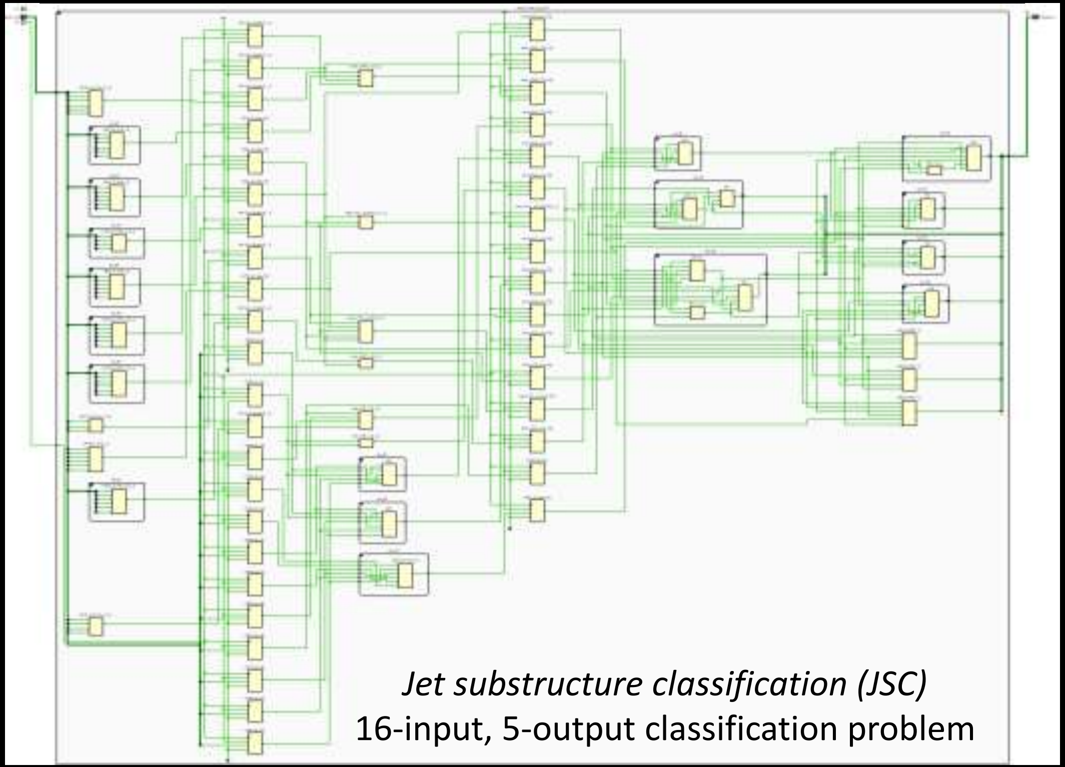
Diversity

LogicNets Results – Tiny (!!!) and Fast

- **DNN in similar area compared to an FPGA 32b adder**

- **High-energy particle physics CERN L1 trigger experiment**
  - Inference rate: 666M inferences/sec*
  - Latency: 3 nsec
  - Resources: 30 LUTs

A Complete Neural Network @ 70% Accuracy!



Jet substructure classification (JSC)
16-input, 5-output classification problem

Synthesized with Vivado 2019.2; F_{Max} equals inference rate
*max device frequency

Diversity

LogicNets Results

- *Quotation from Petersen et al., Dec 2022 @ NeurIPS:*
  - *“FINN […] the **fastest method** for classifying MNIST at an accuracy of 98.4%,”**

	Acc. [%]	LUT	Latency [nsec]	Inferences/sec
FINN	98.4	83k	2,440	1.6M
	95.8	91k	310	12.4M

2x
8x

64x
34x

323x
37x

	Acc. [%]	LUT	Latency [nsec]	Inferences/sec
LogicNets-M	**97.7**	**45k**	**38**	**517M**
LogicNets-S	**95.8**	**12k**	**9**	**458M**

“World’s fastest MNIST classifier”* - now even faster

Synthesized with Vivado 2019.2; F_{Max} equals inference rate

*Petersen et al. "Deep Differentiable Logic Gate Networks." NeurIPS, 2022.

FINN: Diverse Engagements and Open-Source Adoption

- Communications
- Medical
- Sensor Intelligence
- Automotive
- High-energy particle physics
- Aerospace & Defense
- High-frequency Trading

- Open-source Adoption
  - ~2000 stars, 230k+ Brevitas downloads, 72k+ QONNX, 17k+ FINN compiler downloads
- Three best paper awards
- > 1000 citations

Available: Customer support through AMD CSE organization

https://xilinx.github.io/finn

https://github.com/Xilinx/brevitas

Summary

Pervasive AI: dynamic and diverse long tail of AI applications

Paradigm shift towards energy efficiency

Enabling Rapid Specialization with Adaptive Compute Fabrics, Customized Execution Architectures and Agile AI Stacks

Adaptive computing available in great diversity and can help by **customization of hardware execution architectures**

- Dataflow, shrinking precision, fine granular sparsity

Speed-up and automate specialization through graph compilers such as FINN and training libraries Brevitas

Proof points from FINN, Brevitas and LogicNets **demonstrate the potential for energy savings, and addressing truly diverse requirements**

*Petersen et al. "Deep Differentiable Logic Gate Networks." NeurIPS, 2022.

Abstract

- In the context of AI, we face a plethora of challenges that extend beyond the widely discussed performance scalability required to meet the growing demands of compute and storage in the latest models. These challenges encompass sustainability, pervasiveness, agility, and diversity, all of which are needed to cater to a constantly evolving range of applications and algorithms from endpoint to edge and cloud. In this talk, we explore how AMD adaptive devices and agile compiler stacks can provide solutions by delivering post-production hardware specialization and co-designed algorithms. This results in highly optimized AI systems which not only provide the necessary performance scalability but also bring a reduction in carbon footprint while addressing the needs of a broad range of diverse applications with the necessary agility.