



Specialization in Hardware Architectures for Deep Learning

Michaela Blott
Distinguished Engineer
June 2021

Background

> Xilinx

- >> Fabless semiconductor company, founded in Silicon Valley in 1984
- >> Invented the FPGA

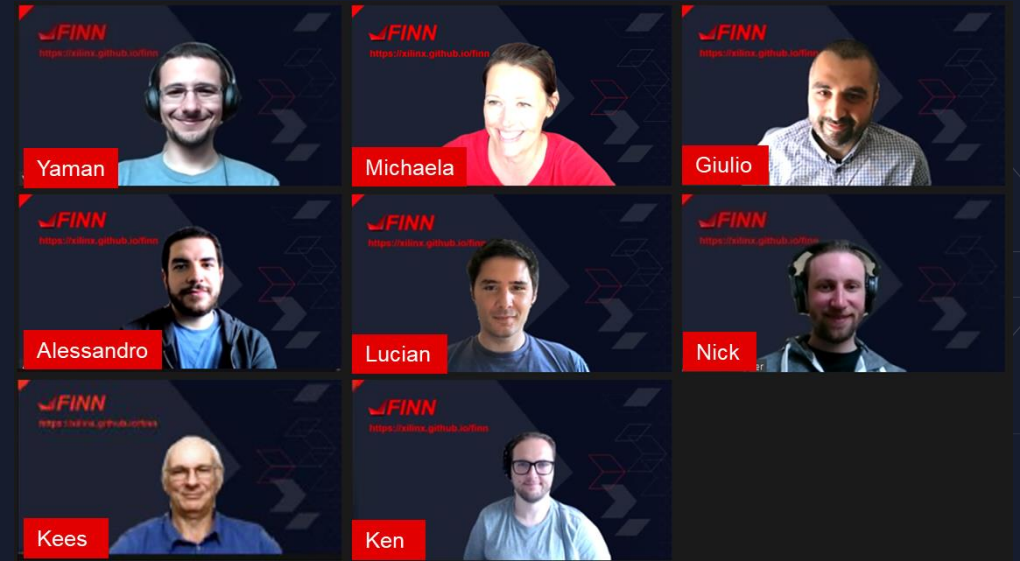
> Xilinx Research Dublin

- >> ~10 researchers plus university program
- >> Plus 4-6 interns typically

> Focus: FPGAs in Machine Learning

- >> Building systems, architectural exploration, algorithmic optimizations, benchmarking


> In collaboration with partners, customers and universities



Lucian Petrica, Giulio Gambardella, Alessandro Pappalardo, Ken O'Brien, Nick Fraser, Yaman Umuroglu ,
Michaela Blott + Kees Vissers

What are FPGAs?

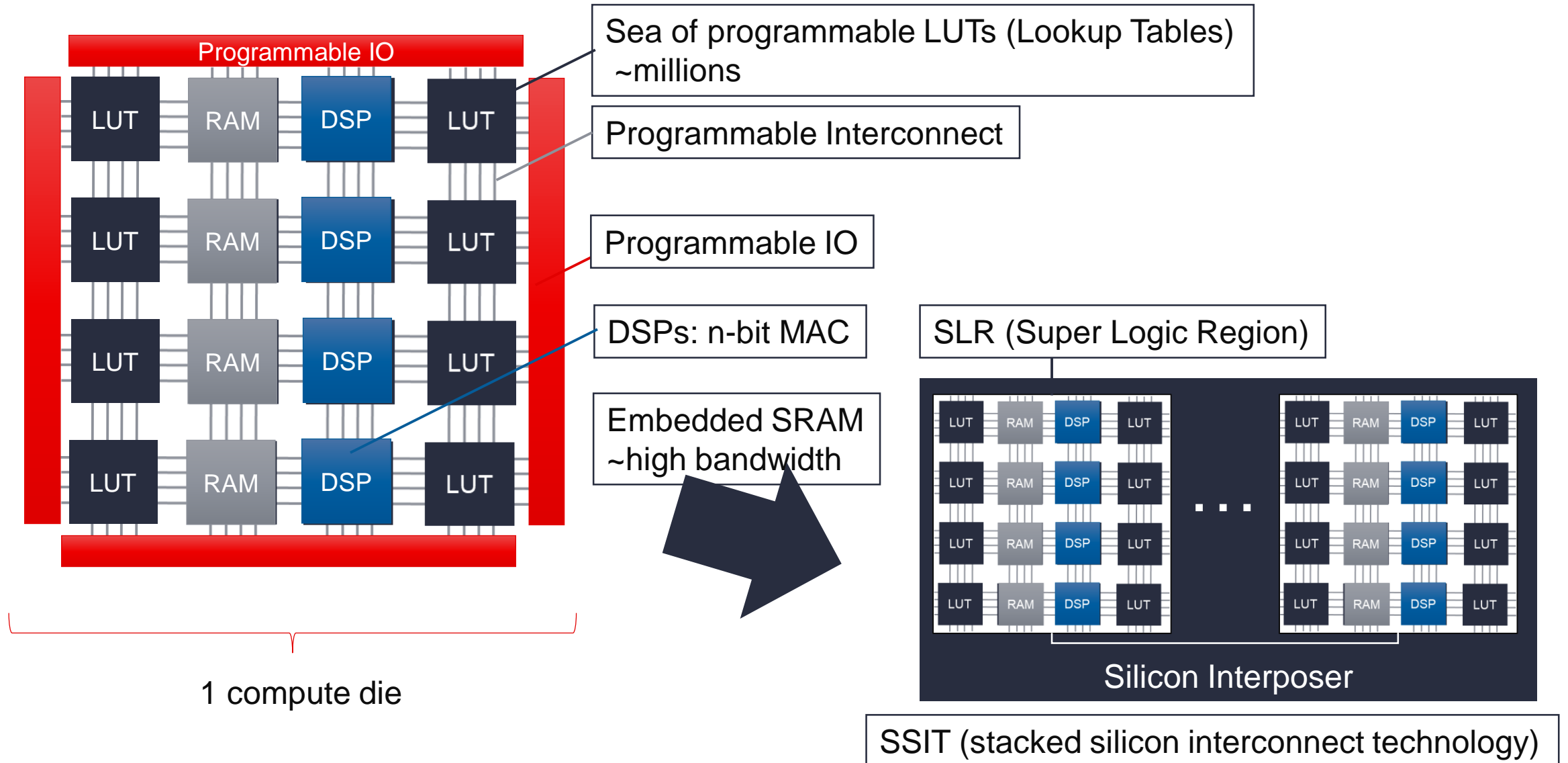
Customizable, Programmable Hardware Architectures

- ▶ The **chameleon** amongst the semiconductors... 
 - Customizes IO interfaces, compute architectures, memory subsystems to meet the application
- ▶ **Use case:** Nothing else works, and you want to avoid ASIC implementation; or ASIC emulation

- Non-standard IOs →
- Different functionality? →
- Higher performance or efficiency metrics? →

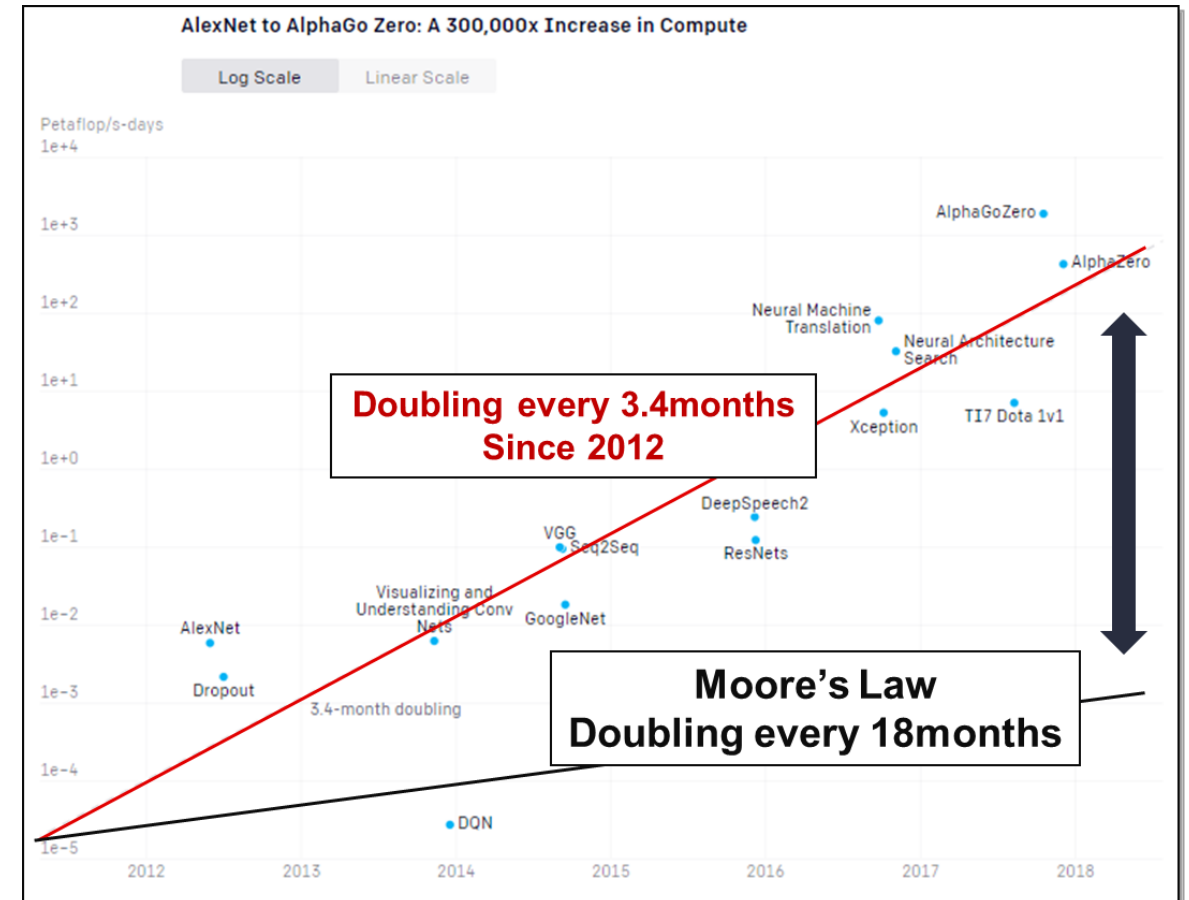


What are FPGAs?

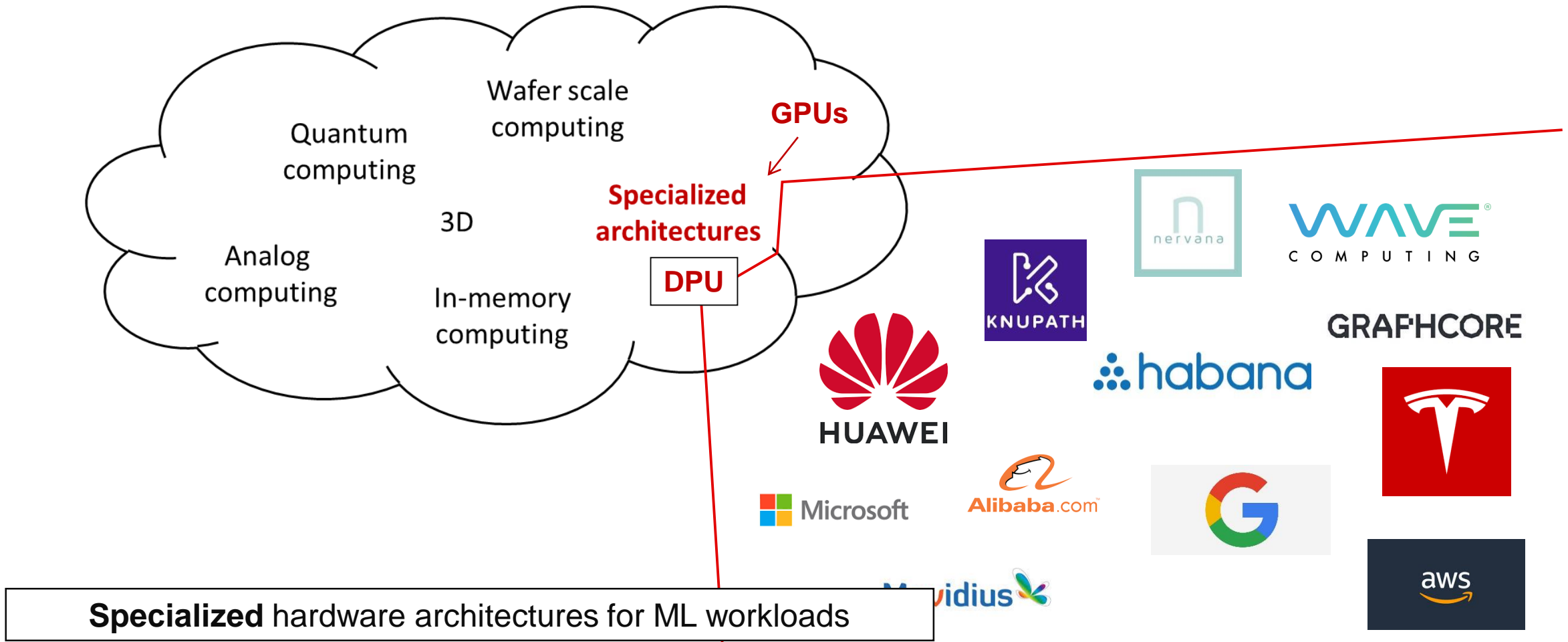


Why do we need specialization in hardware architectures for Deep Learning?

- ▶ DNNs bring huge potential and are penetrating many applications
- ▶ Associated compute and memory requirements are huge
- ▶ Compute requirements are outpacing Moore's Law
- ▶ Hitting the physical limits of silicon-based computing
- ▶ Architectural innovation needed



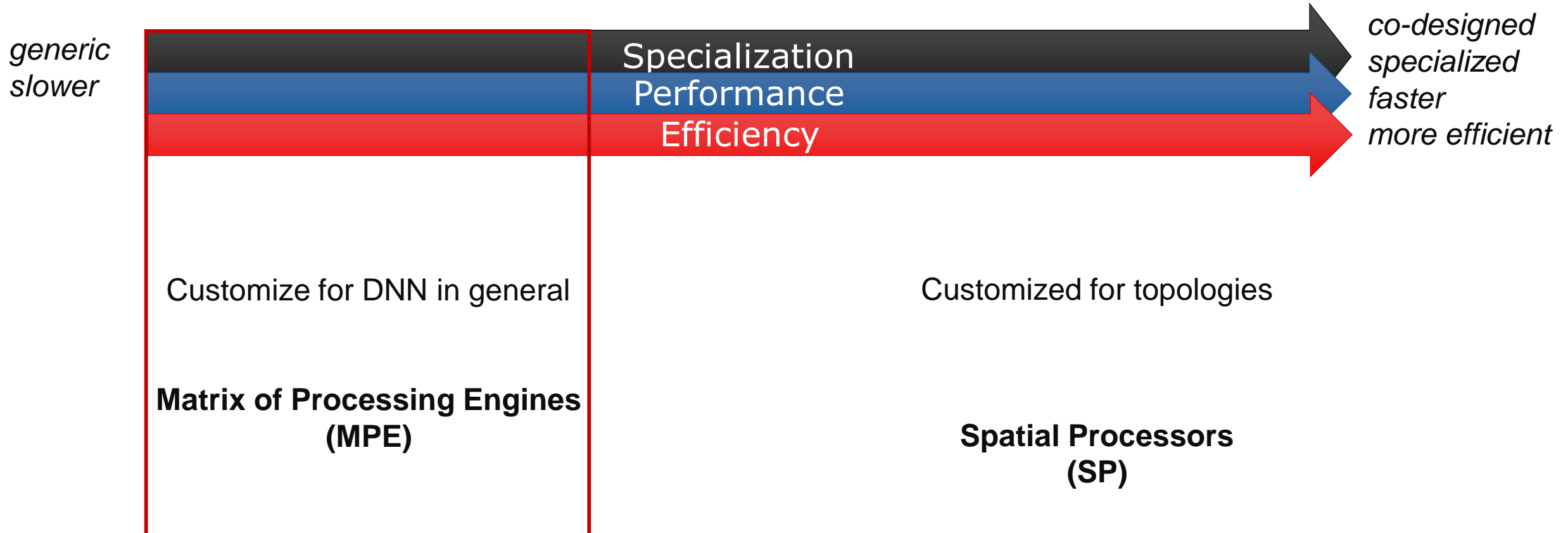
Explosion of Innovative Approaches





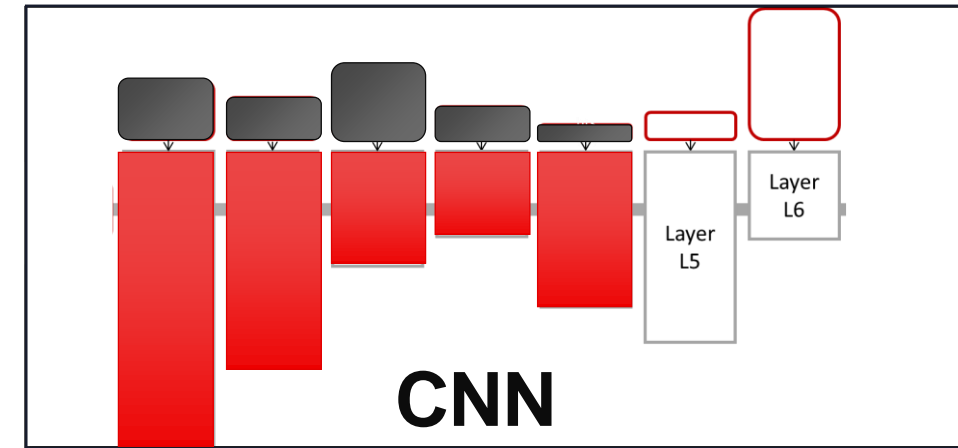
Deep Learning Processor Architectures

Specialization, Performance & Flexibility

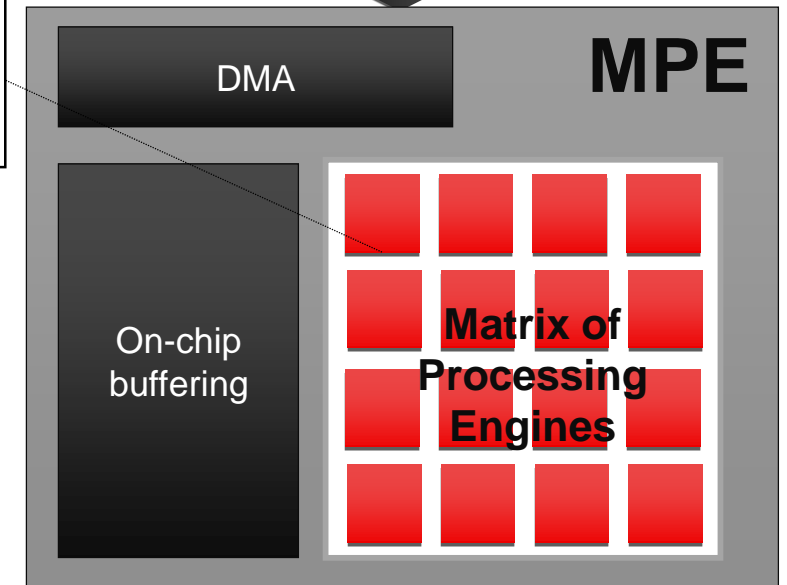
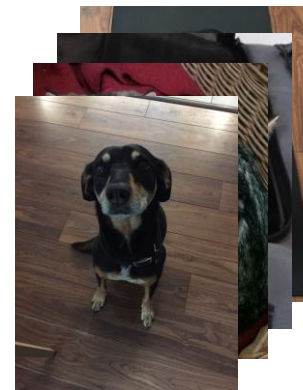


Matrix of Processing Engines Customizing for DNN in General

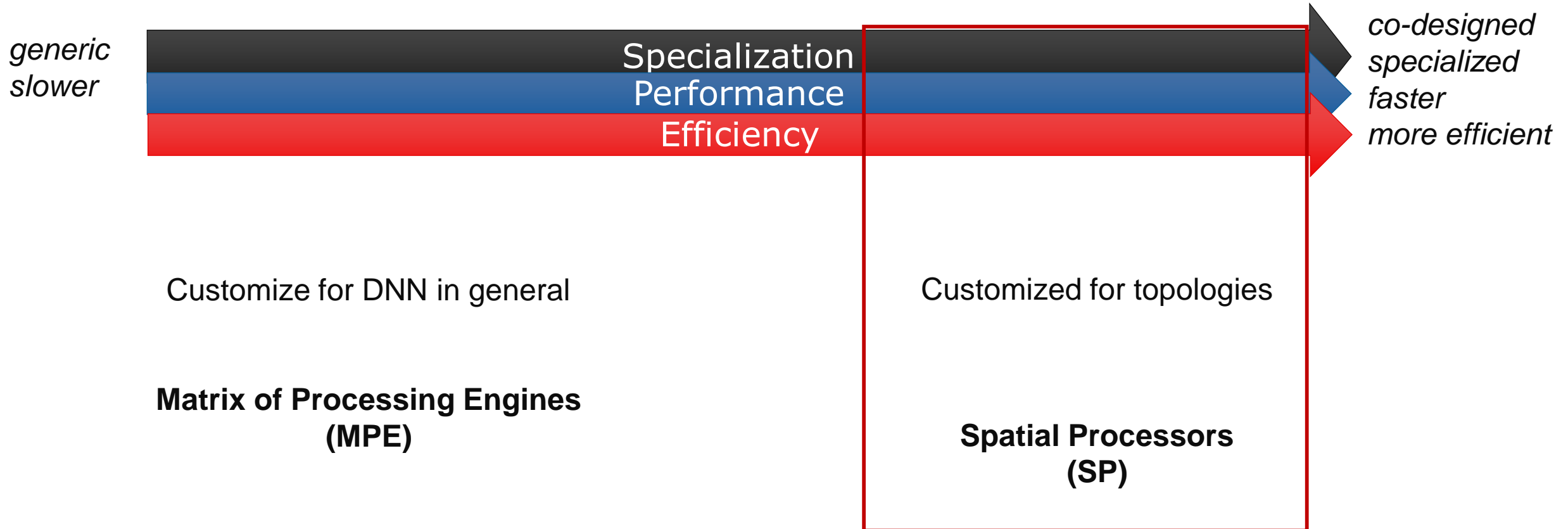
- ▶ Popular layer-by-layer compute
- ▶ Batching to achieve high compute efficiency
- ▶ Customized for ML in general
- ▶ Specialized processing engines
 - Operators
 - ALU types
 - tensor-, matrix- or vector-based



MAC, VLIW,
Vector Processor

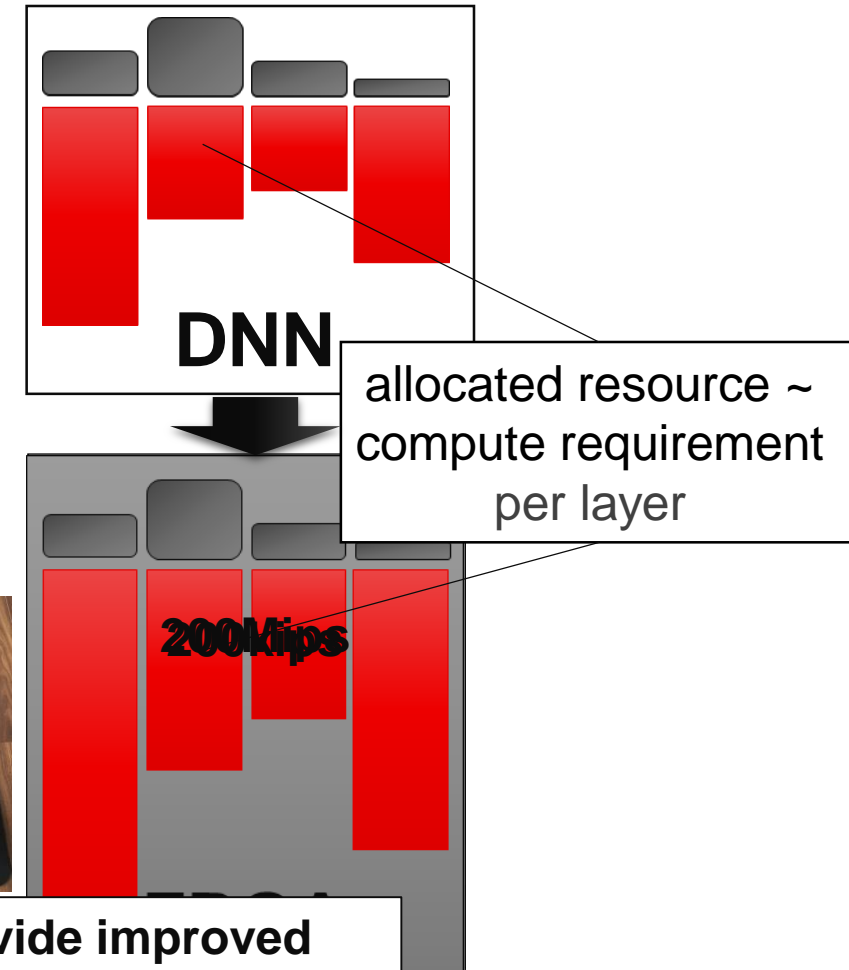


Specialization, Performance & Flexibility



Spatial Processors (SP): Customizing for Specific Topologies

- ▶ Hardware instantiates the topology as a dataflow architecture
 - Customize everything to the **specifics of the given DNN**, any operation, any connectivity
- ▶ Benefits:
 - Improved efficiency
 - Low fixed latency
- ▶ Scale performance & resources to meet the application requirements
 - If resources allow, we can completely unfold to create a circuit that inferences at clock speed and thereby meet these new throughput requirements

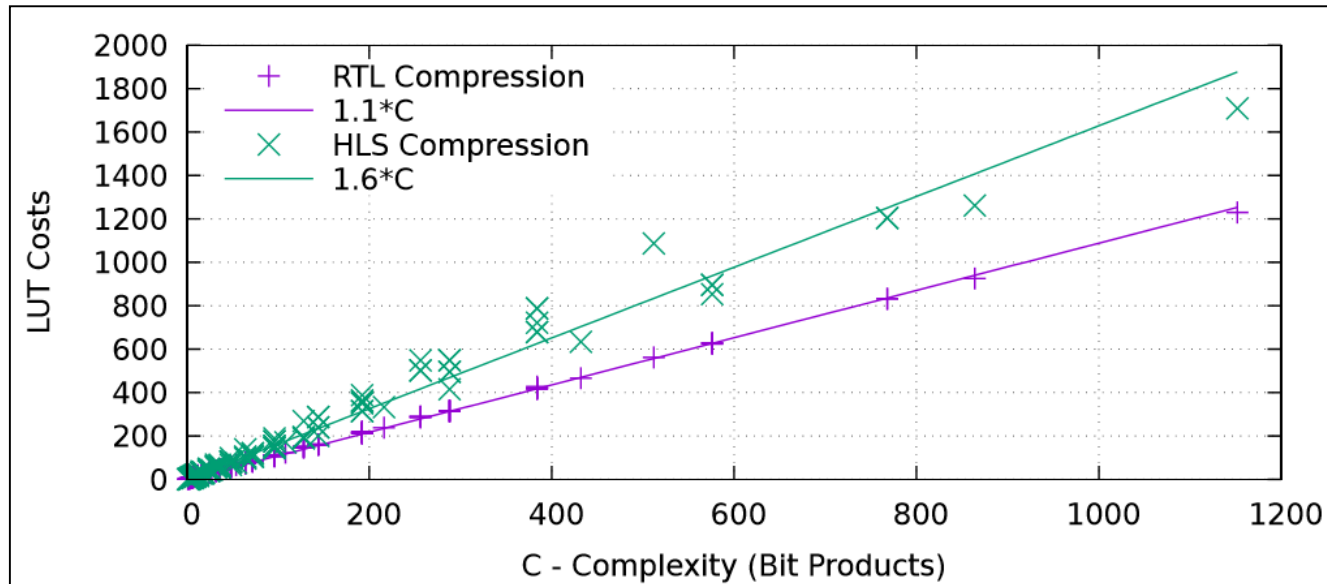


SPs can scale performance, reduce latency and provide improved efficiency

Customizing Arithmetic

Customizing Arithmetic to Minimum Precision Required

- ▶ Shrinks hardware cost & scales performance
 - Instantiate n-times more compute within the same fabric, thereby scale performance n-times
 - **8b/8b -> 1b/1b, RTL => 70x**

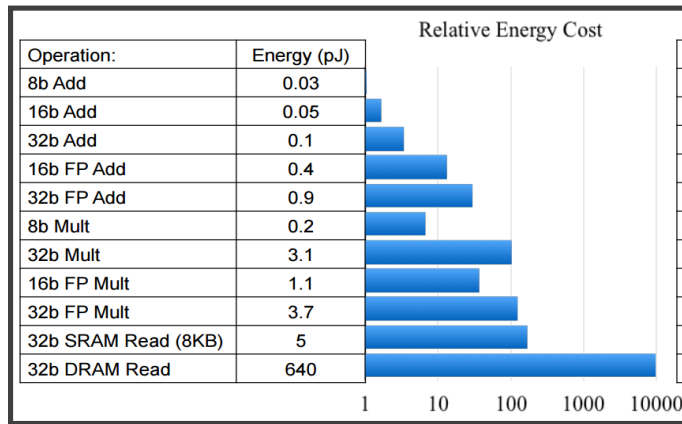


C= size of accumulator * size of weight * size of activation

Customizing Arithmetic to Minimum Precision Required

- ▶ Potential to reduce memory footprint and avoid memory bottleneck
 - DNN inference is typically memory bound
 - DNN model can stay on-chip
- ▶ Inherently saves power

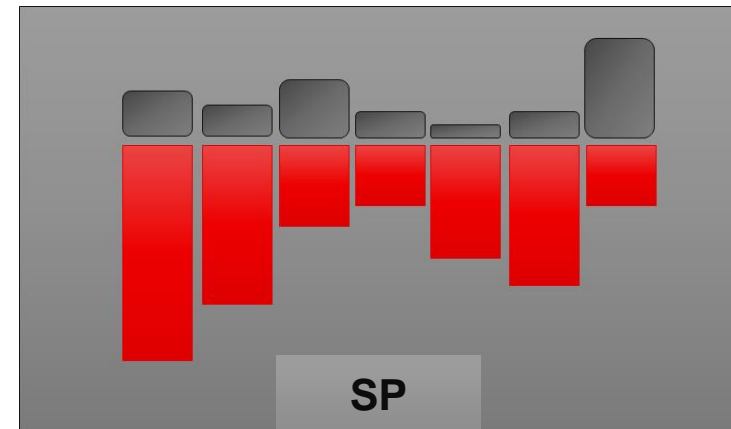
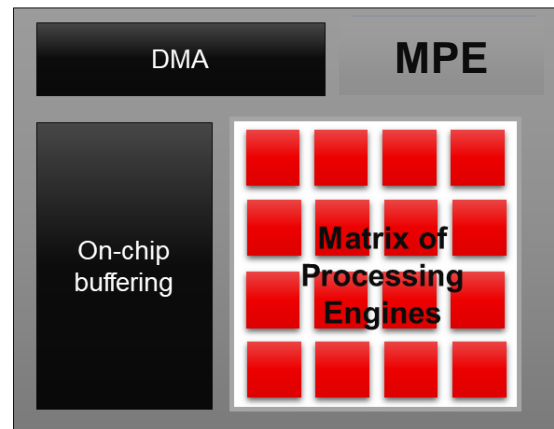
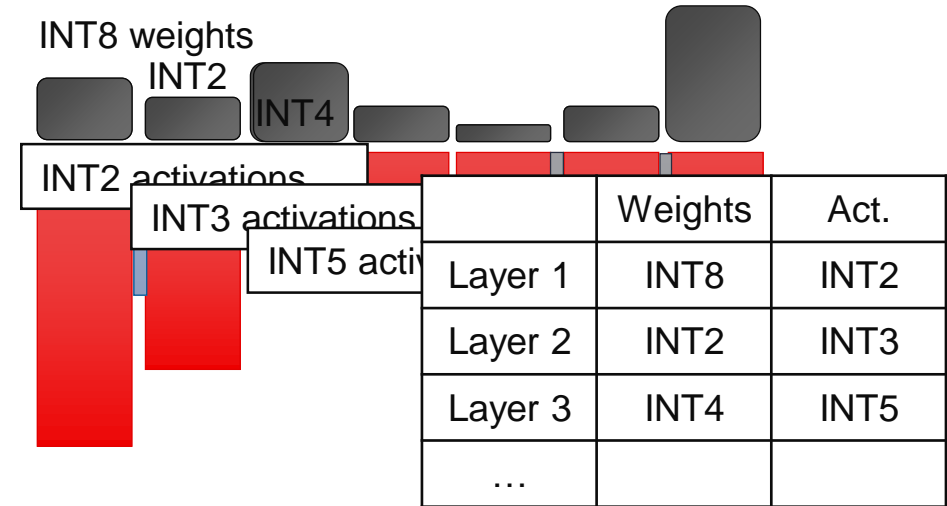
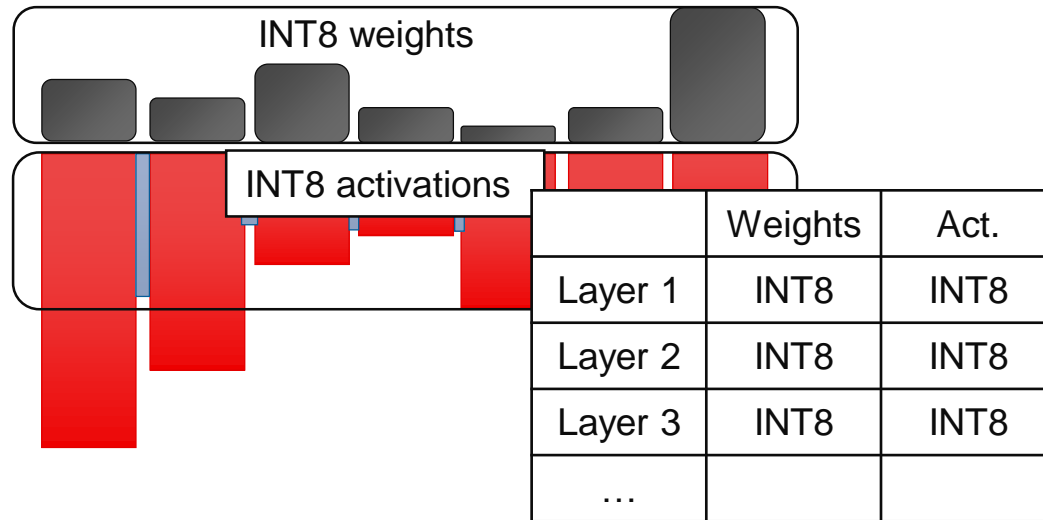
Precision	Modelsize Mbyte (ResNet50)
1b	3.2
8b	25.5
32b	102.5



[Adapted from Horowitz. *Computing's Energy Problem (and what we can do about it)*, ISSCC'14]

Customized arithmetic brings performance, resource, memory and energy benefits
Requires co-design (retraining of CNNs)

Granularity of Customizing Arithmetic



Spatial architectures can exploit custom arithmetic at a greater degree

Challenge

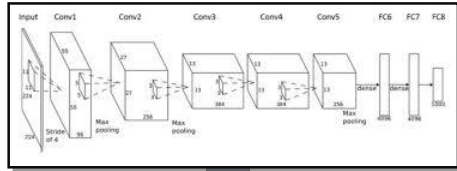
How can we enable a broader spectrum of end-users to be able to specialize hardware architectures and co-design solutions?

Project Mission



- ▶ Providing tools and platforms for exploration of CNN compute architectures
- ▶ End-to-end flow
 - ML engineers can create specialized hardware architectures on an FPGA
 - with spatial architectures and custom precision
- ▶ Open source <https://xilinx.github.io/finn>
 - Transparency and flexibility for the fast changing landscape of algorithms
 - if not supported, you can add your own

From CNN to FPGA Deployment



Brevitas
Training in PyTorch
algorithmic optimizations

- Train a quantized neural network in **PYTORCH**

**ONNX Intermediate Representation
(Reduced Precision DNN)**

FINN compiler
Specializations of
hardware architecture

- Perform optimizations
- Map to Vivado/Vitis HLS
- Create DNN hardware IP with AXI stream interfaces

Deployment

- Integrate generated IP into a larger design
- Support for numerous embedded and server-class platforms



Many Use Cases, Platforms, Datasets and Topologies

▶ Many embedded and server-class platforms

- Multi-FPGA and single-node



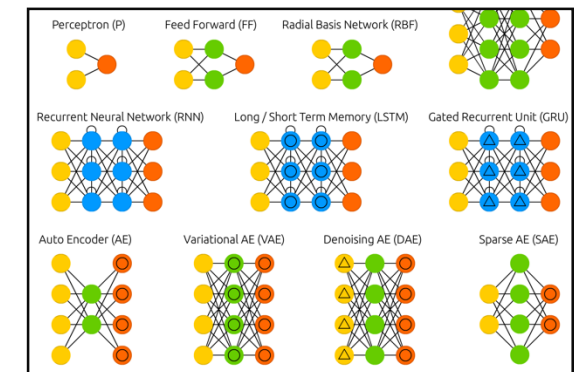
▶ Many more applications

- Radio modulation classification
- Speech recognition
- Facemask detection
- Object recognition with prosthetic hands
- Optical character recognition
- Playing card for solitaire playing robot arm

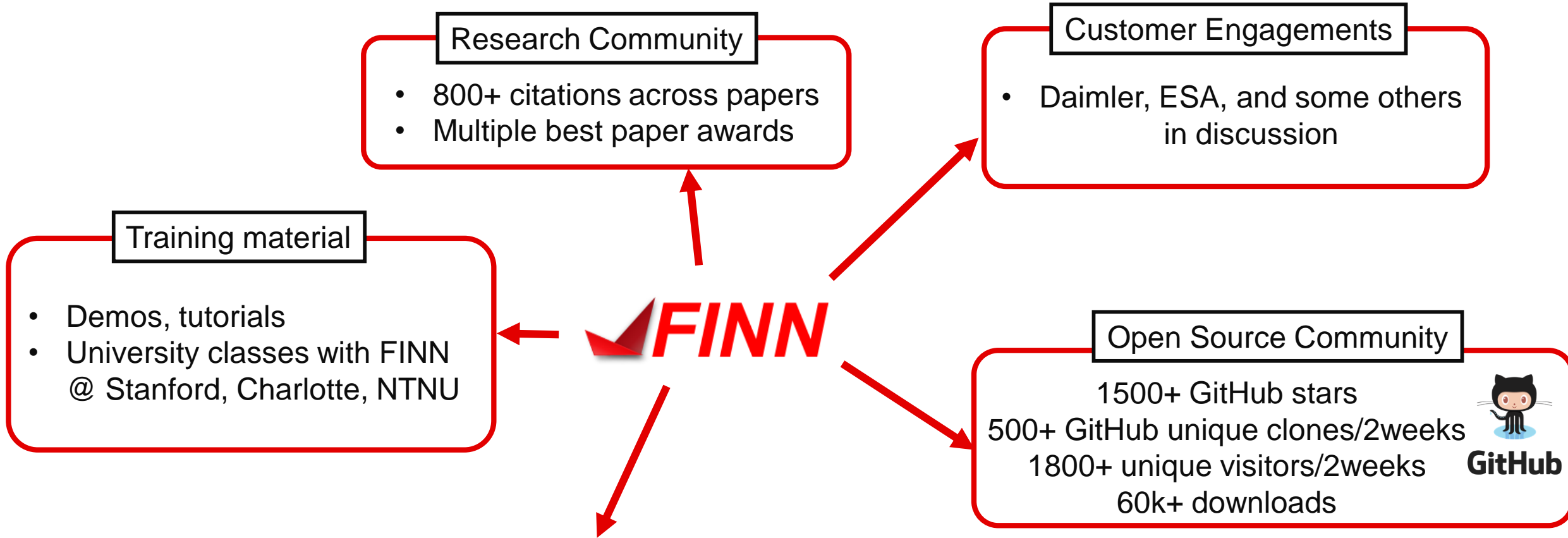


▶ Many topologies

- MLPs, CNV, Yolo variants
- MobileNetv1 & ResNet50
- LSTM
- QuartzNet in progress



Status & Results

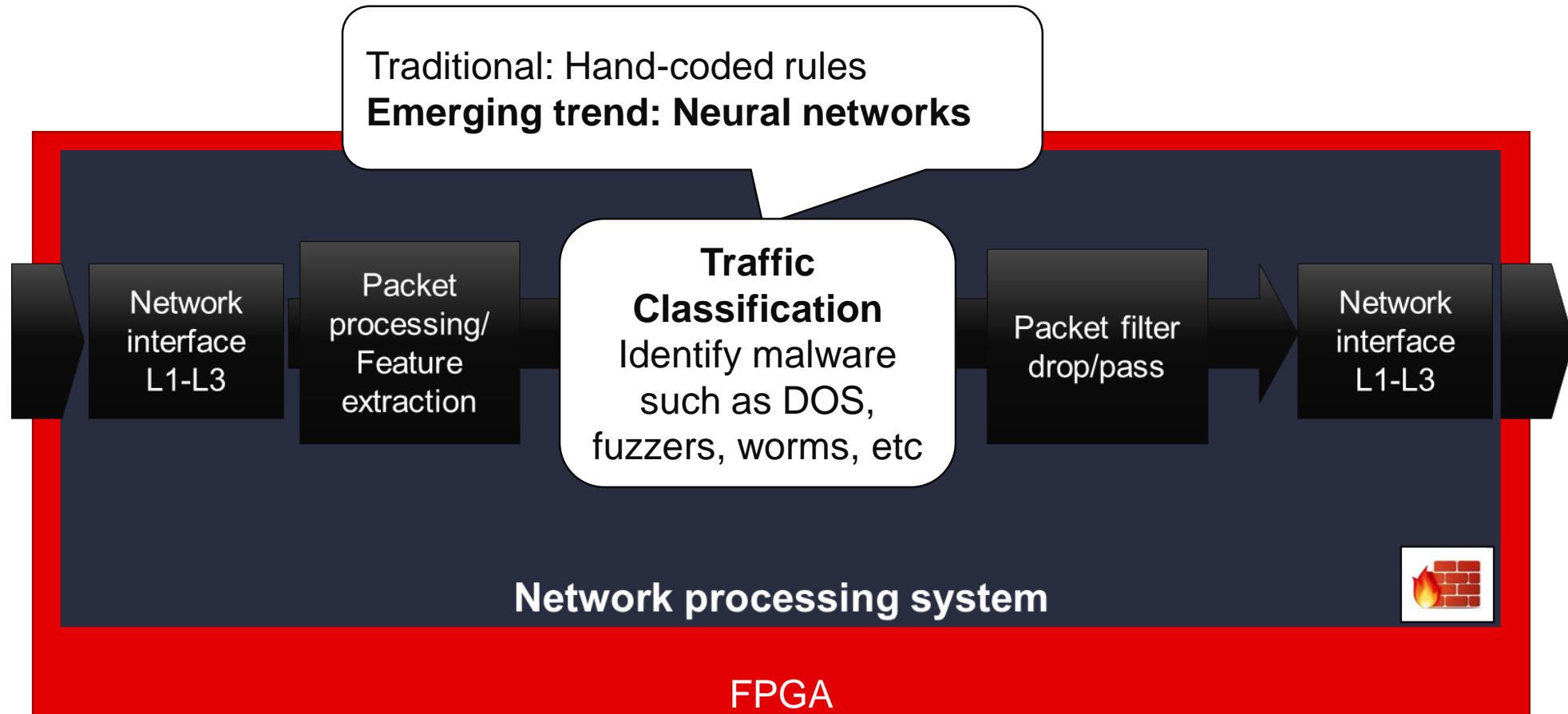


Looking to grow community and build-up industrial applications
If you like to collaborate- we'd love to hear from you 😊

Results



Deep Network Intrusion Detection System



Goal: Implement NN-based traffic classifier delivering 100G **line-rate** throughput = 150 Mips
Latency sensitive (buffer 10s of MB/msec)

[1] Moustafa, Nour, and Jim Slay. [UNSW-NB15: a comprehensive data set for network intrusion detection systems \(UNSW-NB15 network data set\).](#) Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015.

[2] Murovič, Tadej, and Andrej Trost. "Massively parallel combinational binary neural networks for edge processing." *Elektrotehnikski Vestnik* 86.1/2 (2019): 47-53.

Results



	MPE		FINN
	Vitis AI	Brevitas →	FINN (fold 1)
Topology / #layers / #OPs	MLP / 3 / 92KOPs	Same DNN topology Reduced precision Good Accuracy	MLP / 3 / 92KOPs
Datatype	8bit		2bit
Accuracy	92.3%		91.9%
Performance		FINN compiler →	
Throughput	22kips	1000x throughput 1000x latency reduction Meets comms requirements	300Mips
Latency (compute only)	26us		18ns
Resources		→	
Compute (KLUTs, DSPs)	122,1124	Low resource footprint (especially memory) Low clock rate	~8 – 69, 0 *
Memory (BRAM, URAM)	290, 92		0, 0
Clock	300/600MHz		300MHz

- **>1000x performance improvement over Vitis AI, less resources,**
- **100Gbps line rate (150Mips)**
- **Exploits: dataflow processing, reduced precision, fine-grained sparsity**

Summary



Summary

- ▶ Spectrum of innovative architectures emerge to address upcoming compute and memory requirements in DNNs
- ▶ Specialization of hardware architecture are critical to scaling architectures
 - In particular for extreme throughput applications as we see for example in communications
- ▶ We looked at the NIDS example which showed the tremendous benefits we get from quantization and spatial implementations

Infrastructure for Experimentation & Collaboration

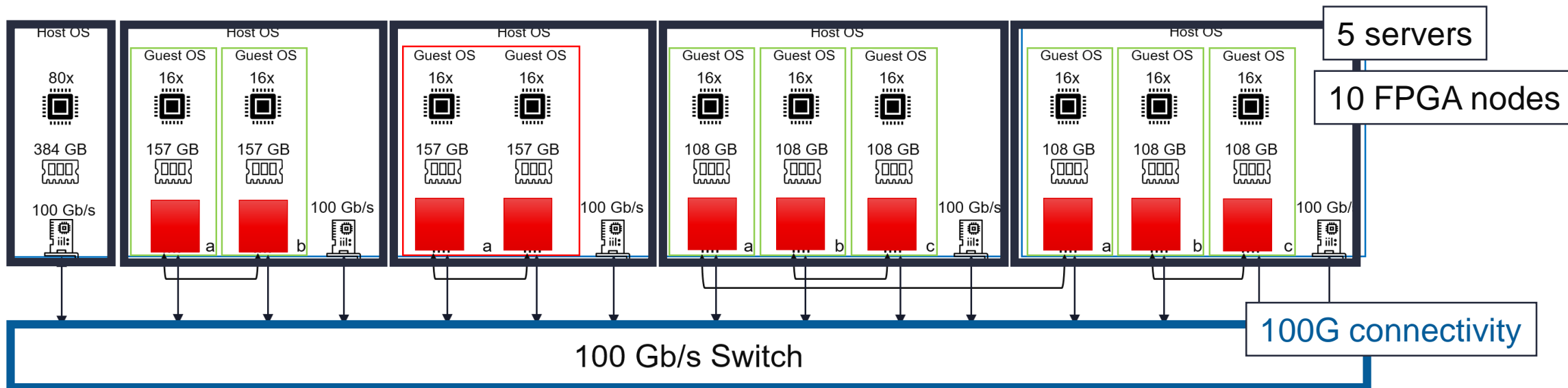
- ▶ Xilinx academic compute clusters (XACC)

- 4 centres world-wide
- Free to use
- Enabling research community
- Explore innovative compute architectures



- ▶ Flexibility, networked FPGAs

- ▶ Many examples emerging: <https://xilinx.github.io/xacc/>





Thank You

