Meeting the Critical Needs of Accuracy, Performance and Adaptability in Embedded Neural Networks

Aman Sikka

Chief Architect

Quadric

Quadric

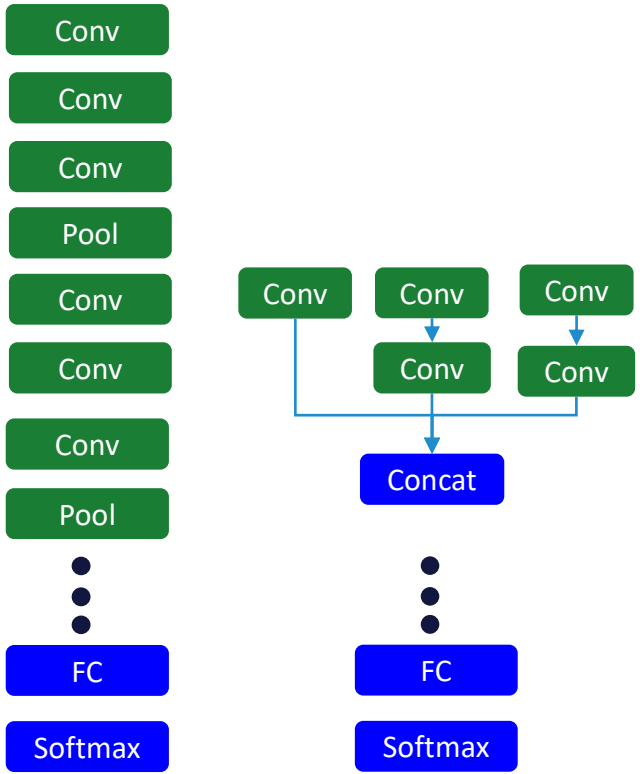
Agenda

- Challenges in current NPUs (Neural Processing Units)
  - Trends in neural networks
  - Increase need for DSP like ops but DSP cannot be a fallback
- Back to basics
  - Fixed point vs. floating point
  - Designing a flexible architecture
- Conclusion
- Q/A

Traditional NNs and Hardware

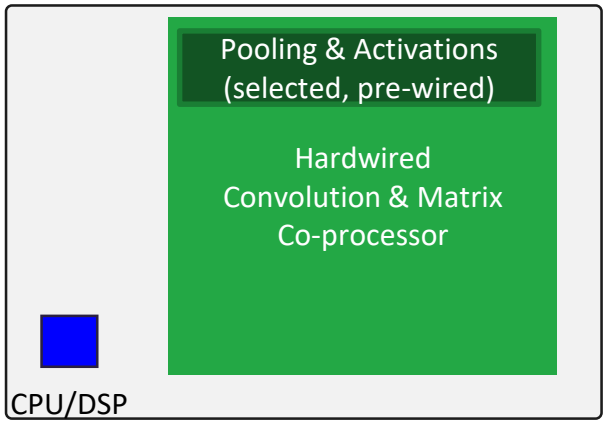
Quadric

Traditional CNN: Mainly MAC dominated

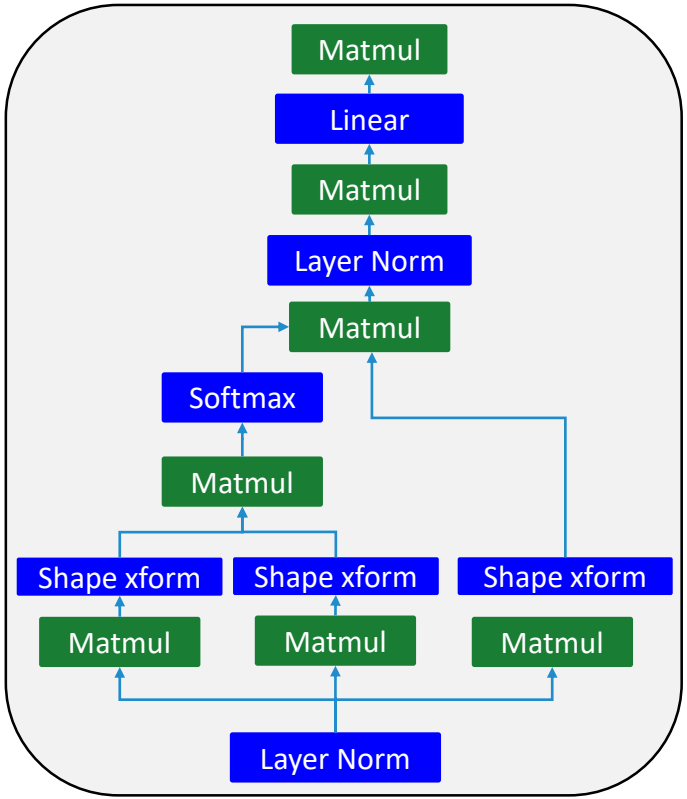


NPU Ops (matrix, pooling, activation)

Classic Algorithm & Control Ops



Transformers: Can be heavy on DSP-like compute



Attention block

- NPU operations (matrix, pooling, activation)
- Classic Algorithm & Control Ops

Every data transfer between NPU block and CPU-DSP decreases performance and adds power

Energy Cost of 32b data element transfer from ALU/MAC to	
Reg File	1
LRM	2-3X
L2 MEM	70X
Off-chip DDR	225X

Non-NPU ops are here to stay

- SWIN network requires ~77% of the workload to be executed on a programmable device

Cadence says:

"When a SWIN network is executed on an AI computational block that includes an AI hardware accelerator designed for an older CNN architecture, only ~23% of the workload might run on the AI hardware accelerator's fixed architecture. In this one instance."

https://www.cadence.com/en_US/home/resources/white-papers/why-a-dsp-is-indispensable-in-the-new-world-of-ai-wp.html

Quadric

Non-NPU ops are here to stay

Data Transformations

- Reshape
- Transpose
- Shifted window
- Patch creation
- Embeddings lookup
- …

Inference

- Softmax
- Layer norm
- Group norm
- Instance norm
- Pixel Co-relation
- Positional encodings
- 2 input matmuls
- Look up tables
- …

Pre/Post Processing

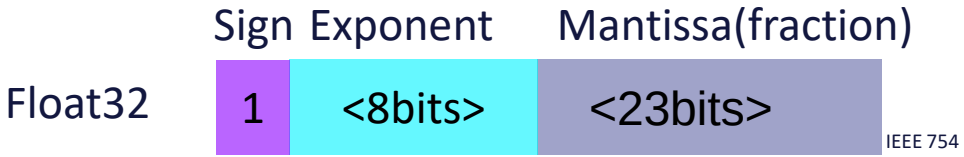
- NMS
- ROIAlign
- Noise reduction
- FFT/RFFT
- Equation solver
- Mean subtract
- …

Where are we headed?

- Floating-point operations are getting more common during inference and can take a large part of compute
- Future designs would have multiple DSP cores, CPUs, AI accelerators, vision accelerators …

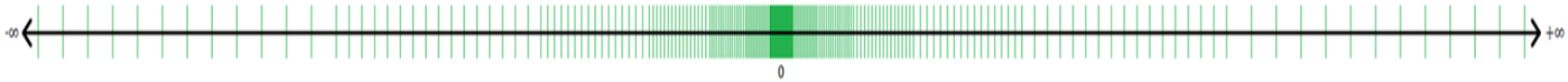
Back to basics: Float32 vs Fixed Point

Float32 representation



$$-1^{s} * 2^{exponent} * mantissa$$

- **Range**: refers to the span of values that can be represented. *Exponent* provides the dynamic range.
- **Precision**: refers to the ability of a format to distinguish between two close values. *Mantissa* provides the precision within a range.



Quadric

What does floating point offer?

- Floating point offers
  - Better dynamic range.
  - Ease of development, as a user doesn’t need to adjust for precision and range
- But at a significant cost – power consumption!

Do we really need floating point?

- **Known ranges of input and output**: In all quantized neural nets the input and output ranges are well known. One can use a calibration dataset to identify all that.
- **Fixed ranged ops**: Sin, cos, sigmoid, softmax, norm, etc.
- **Per operation range estimate**: In almost all neural nets the data ranges across layer and operations can easily be gathered with a calibration data set

Do we really need floating point?

- **Dynamic fixed point** : Based on the operations done internally one can analyze the math and change fixed point precision per calculation.

Fixed point32 representation

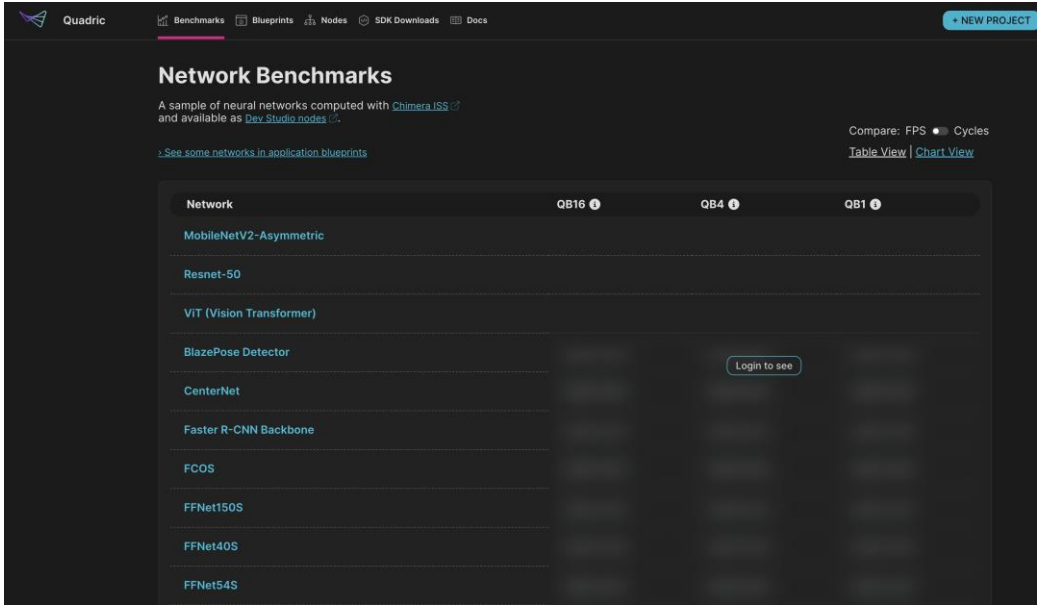
- **Range and Precision**: can be controlled by the developer. Precision can be represented in for 31 fractional bits.

- Example:
  - FixedPoint32<24>.. Base *int32* with *24* bits to represent fractional value
  - FixedPoint16<11>.. Base *int16* with *11* bits to represent fractional value

Fixed point accuracy numbers

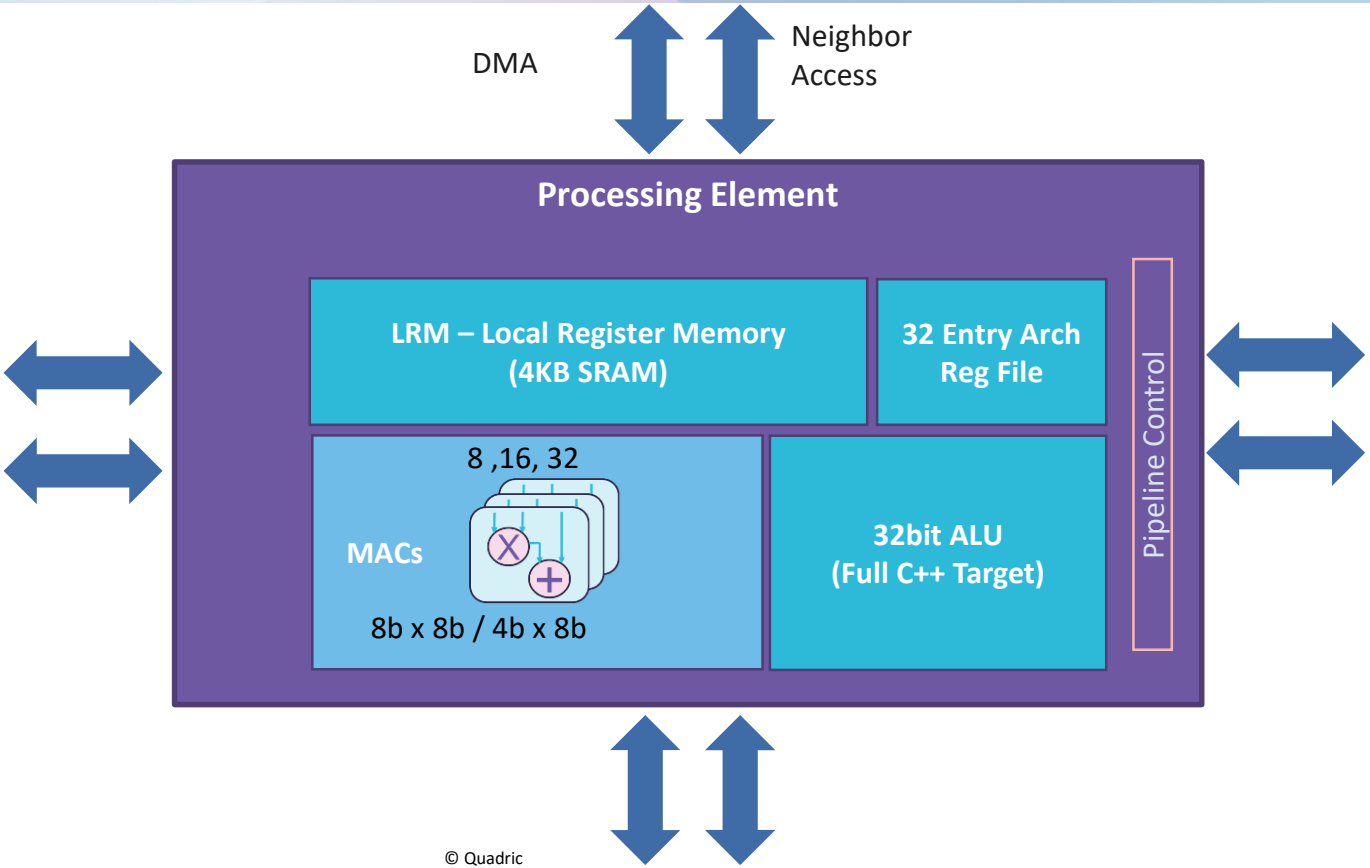
- Quadric already supports 60+ networks (transformers, detector, segmentation, classifier …) within **<1%** top1 accuracy loss compared to floating-point models.

http://quadric.io/evs24

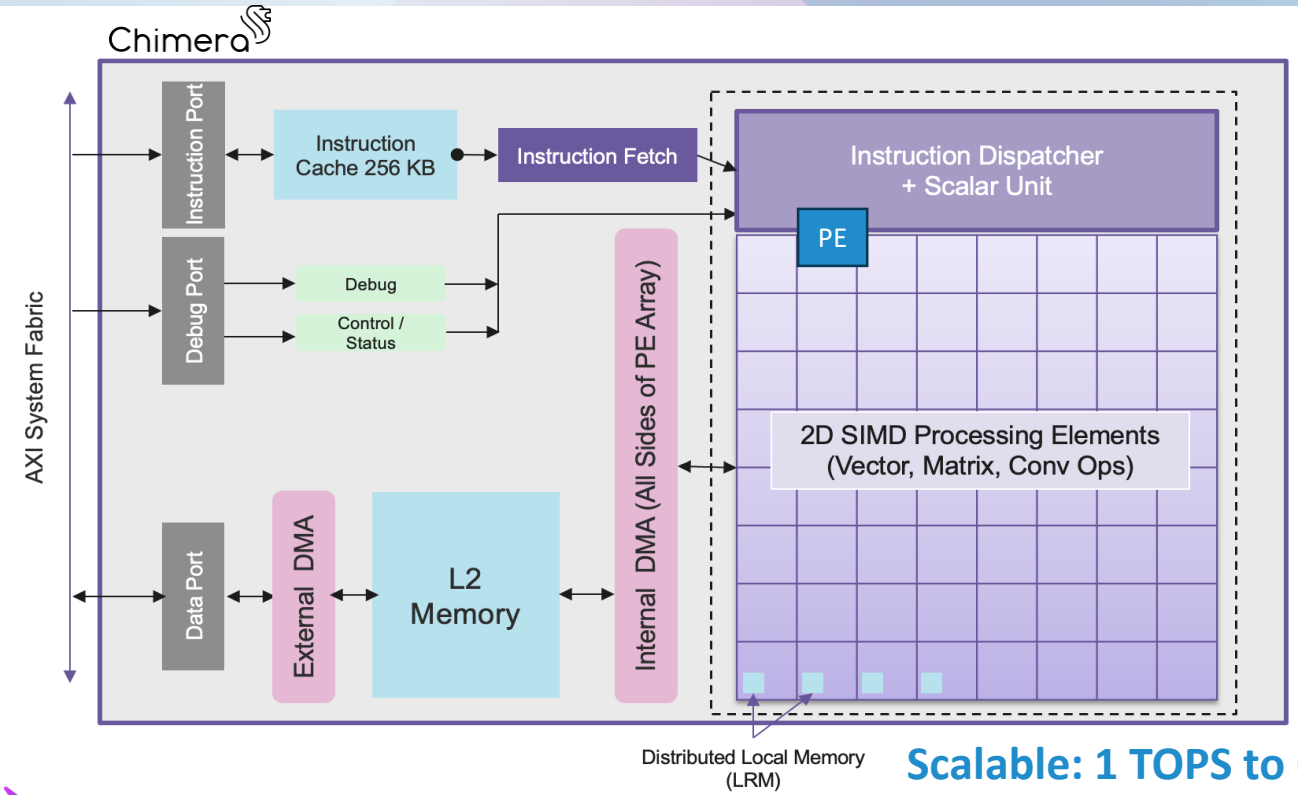


Back to basics: Designing an architecture from lessons learned

Chimera GPNPU

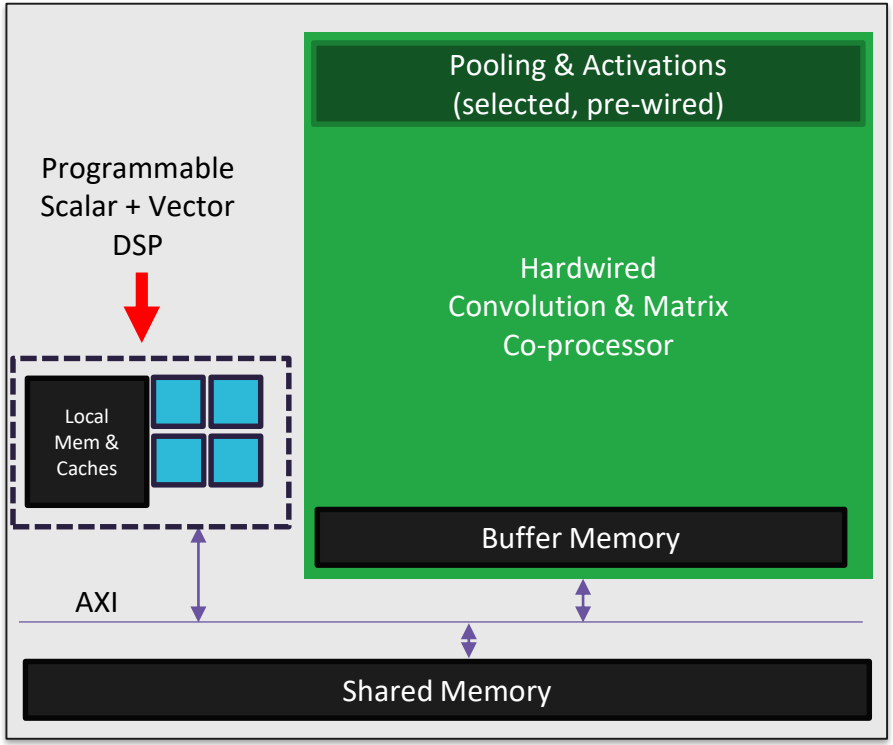


Chimera GPNPU



Scalable: 1 TOPS to 64 TOPS single core
Up to 512 TOPS Multi core

Existing solutions

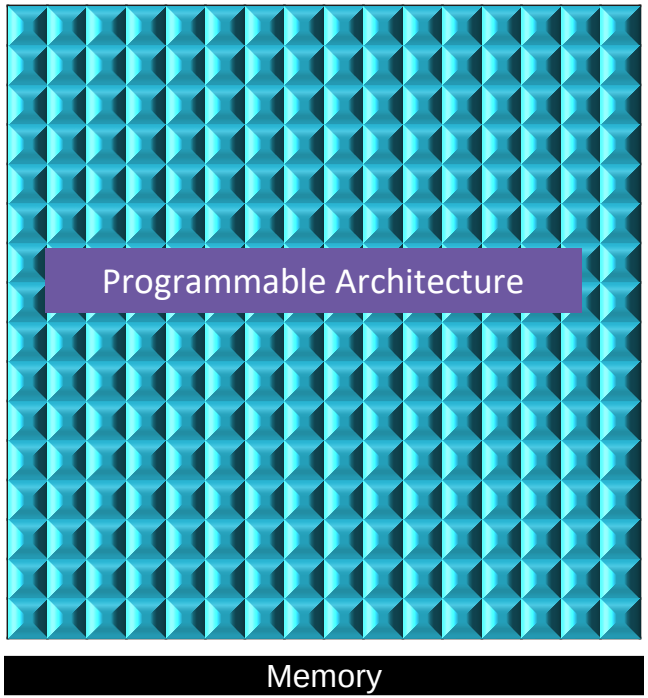


Offers very little programmability

- Code partitioning/ programming complexity
- System complexity / power
- Accelerator brittleness
- No ability to modify hardware after tapeout
- Leads to lower-performance "fallback" onto the DSP or CPU
- Shortens market lifetime of SoC

Quadric

Chimera GPNPU — A code powered AI engine

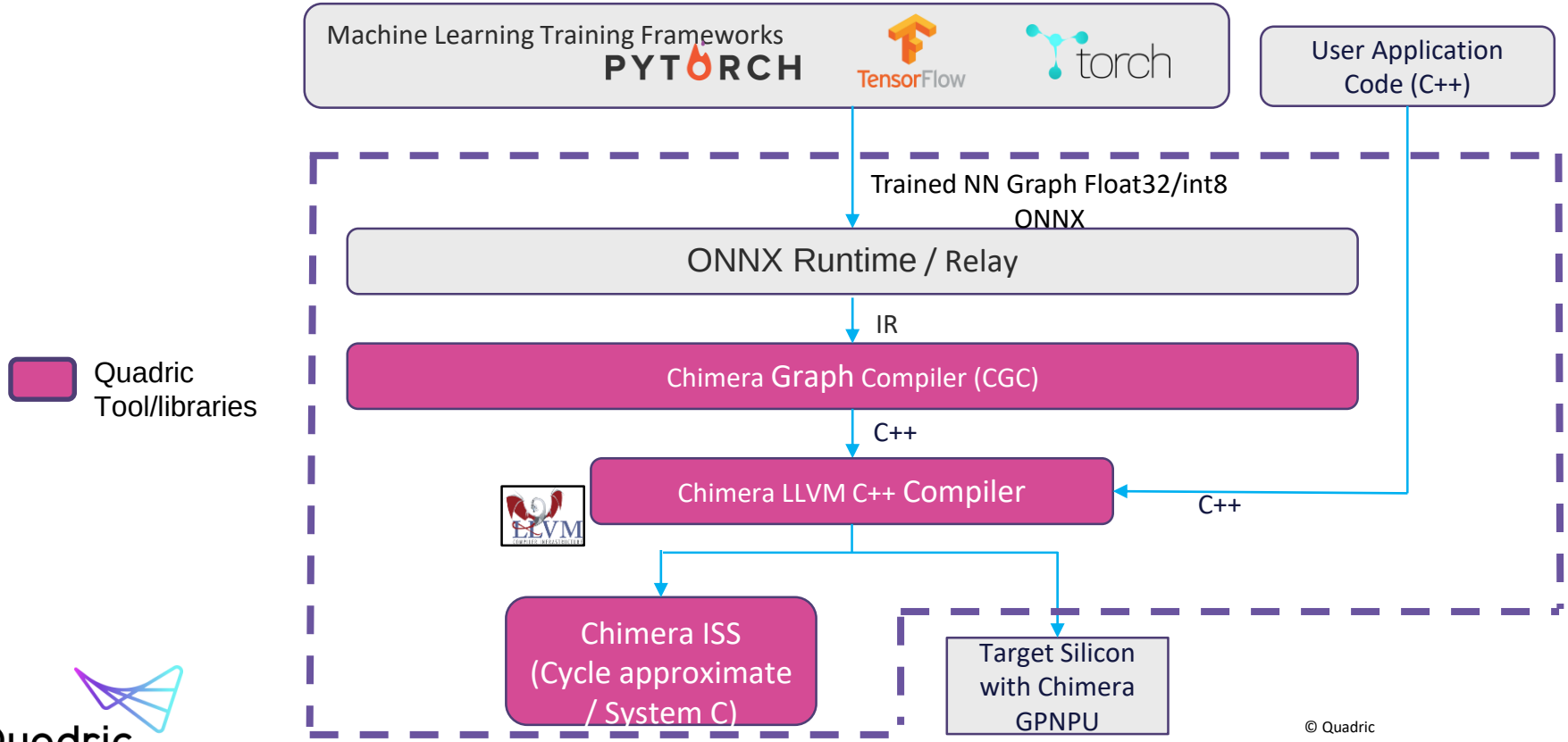


100% of GPNPU is end user programmable

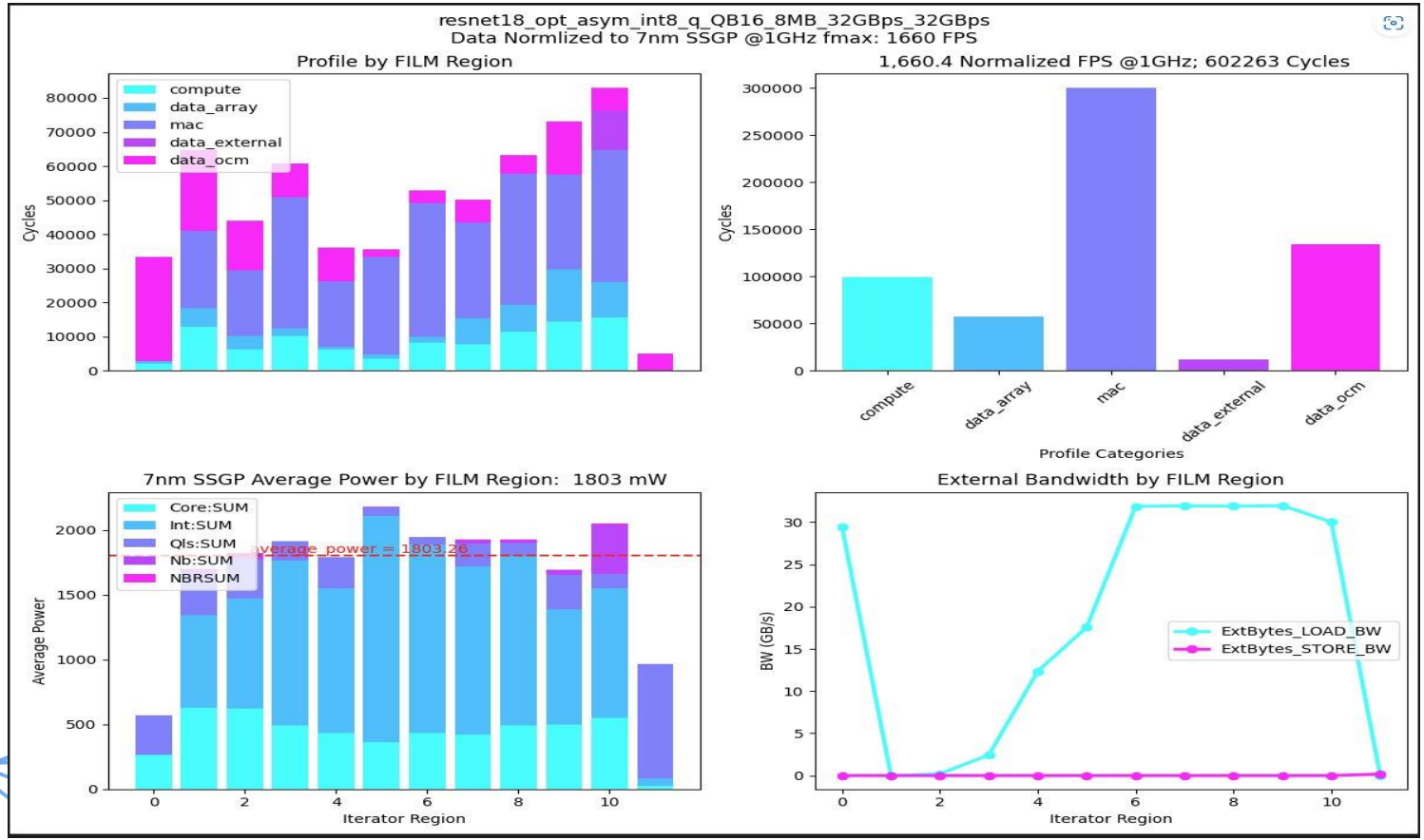
- Dramatically easier software programming model with ability to program in C++/python
- Simpler SOC architecture
- Long SoC lifespan – easy ML operator support

Quadric

Programming model



Instruction-based simulation gives detailed bandwidth, power, and performance insights

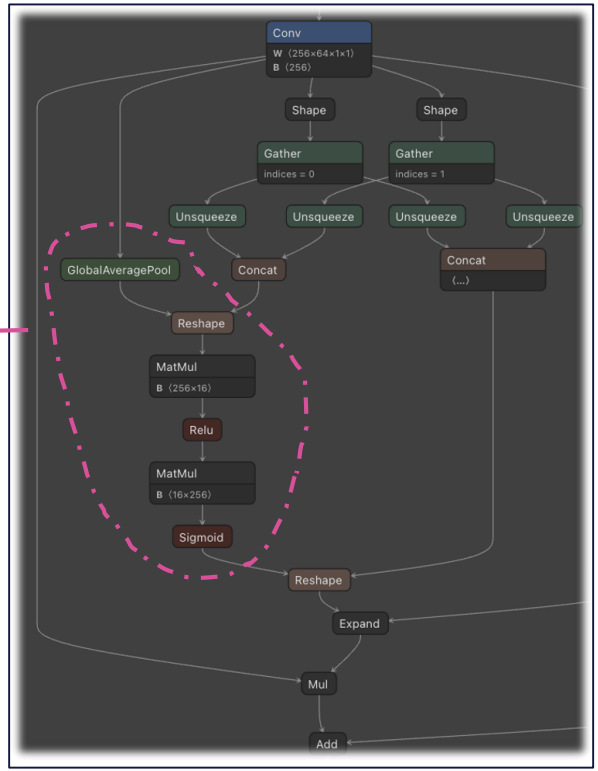


Custom operator support

- Custom implementation of nodes/subgraphs
  - e.g., NMS, proprietary layers, custom operators

```
template <typename OcmWeightsShape>
void CustomOperator(DdrInTensorShape::ptrType ddrInPtrA,
                    DdrOutTensorShape::ptrType ddrOutPtr) {
  /* Operator implementation*/
  .
  .
  .
  .
  .
  .
  .

}
```



Conclusions

Quadric

Conclusions

- Floating-point unit can easily be replaced with fixed-point integer math equally well. Same accuracy with lower power & higher performance.

- Fixed operation units (ASICs) only work in niche applications. In today's world with AI algorithms changing every 3 weeks, one needs a very flexible architecture which is easy to program.

- Operations requiring non-mac compute are becoming very common. Having multiple DSP/special cores is not the right fallback....

Need a unified architecture to handle all workloads…

Q&A

Quadric

About Quadric: http://quadric.io/evs24

embedded VISION SUMMIT®

Pure play Semiconductor IP Licensing

- Processor IP & Software Tools

Edge / device AI/ML Inference + DSP processing

HQ: Silicon Valley – Burlingame CA
Total Venture Capital Raised: $48M

May-2023: First IP delivery, DevStudio Online

Patents: 25 Granted

Visit us at Booth: 717

Silicon Proven Test Chip

Successful silicon in 2021

Quadric