tinyML® Talks

Enabling Ultra-low Power Machine Learning at the Edge

"The Akida Neural Processor: Low Power CNN Inference and Learning at the Edge"

Kristofor Carlson – BrainChip Inc.

September 1, 2020

www.tinyML.org

tinyML Talks Sponsors

Additional Sponsorships available – contact Bette@tinyML.org for info

Deeplite

WE USE AI TO MAKE OTHER AI FASTER, SMALLER AND MORE POWER EFFICIENT

Automatically compress SOTA models like MobileNet to <200KB with **little to no drop in accuracy** for inference on resource-limited MCUs

bit.ly/Deeplite bit.ly/Deeplite

Reduce model optimization trial & error from weeks to days using Deeplite's **design space exploration**

APPLY NOW

Deploy more models to your device without sacrificing performance or battery life with our **easy-to-use software**

VISIT bit.ly/Deeplite FOR MORE INFO

arm mobilityXlab AI 100 CBINSIGHTS

TinyML for all developers

C++ library

Arduino library

WebAssembly

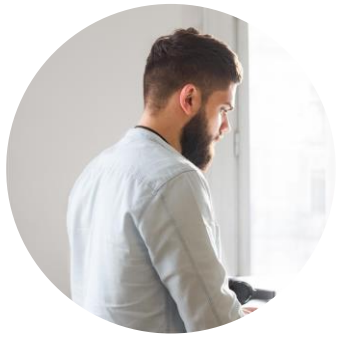
Dataset

Acquire valuable training data securely

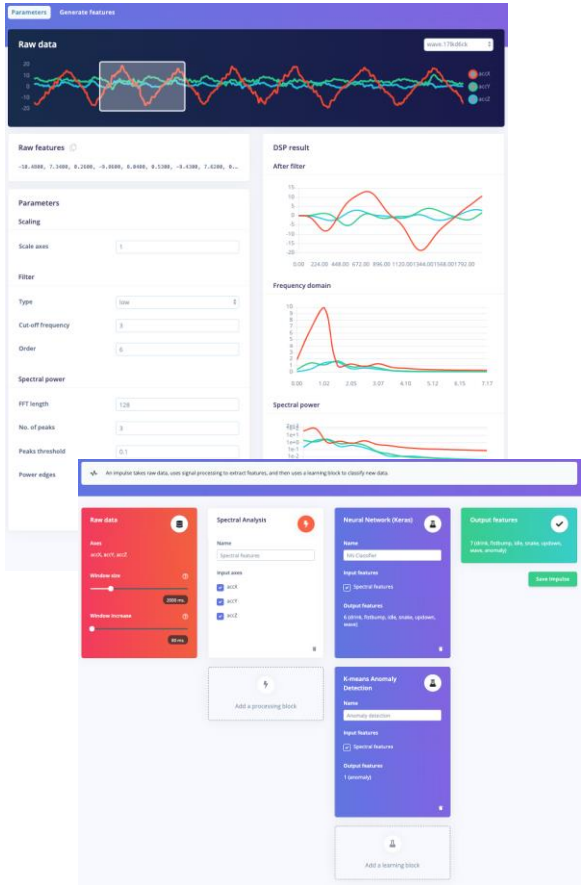
Enrich data and train ML algorithms

Edge Device

Real sensors in real time
Open source SDK



Impulse



Embedded and edge compute deployment options

Test impulse with real-time device data flows

Test

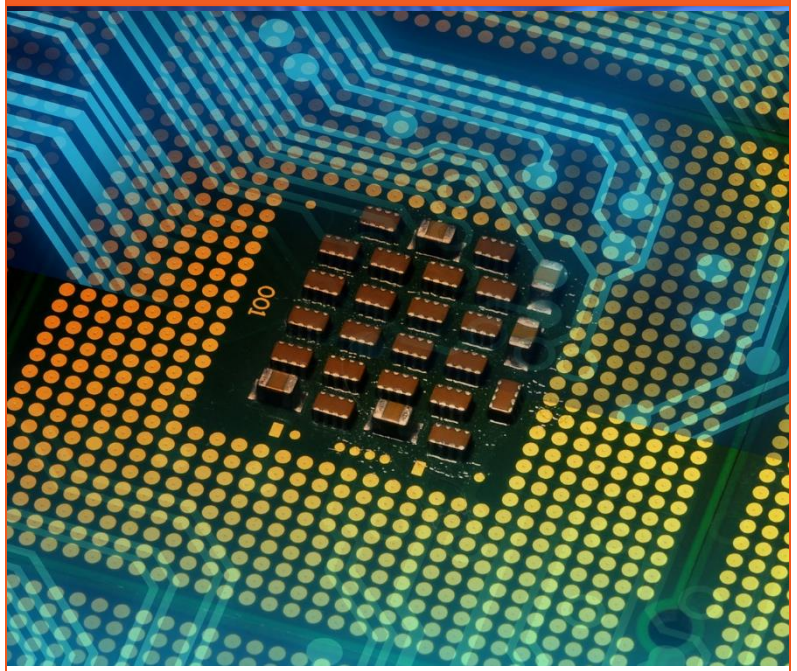
Get your free account at http://edgeimpulse.com

Maxim Integrated: Enabling Edge Intelligence

Sensors and Signal Conditioning

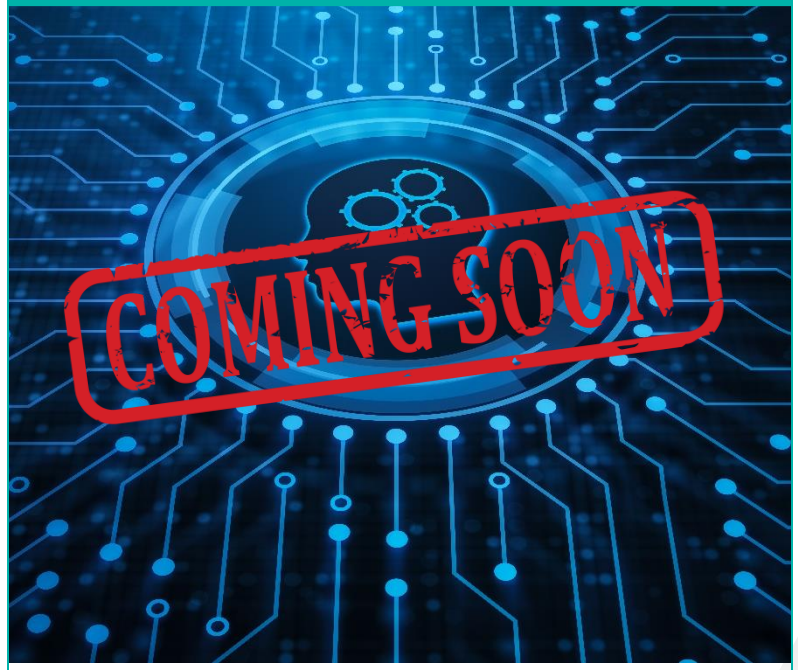
Health sensors measure PPG and ECG signals critical to understanding vital signs. Signal chain products enable measuring even the most sensitive signals.

Low Power Cortex M4 Micros



The biggest (3MB flash and 1MB SRAM) and the smallest (256KB flash and 96KB SRAM) Cortex M4 microcontrollers enable algorithms and neural networks to run at wearable power levels

Advanced AI Acceleration



AI inferences at a cost and power point that makes sense for the edge. Computation capability to give vision to the IoT, without the power cables. *Coming soon!*

Qeexo AutoML for Embedded AI

Automated Machine Learning Platform that builds tinyML solutions for the Edge using sensor data

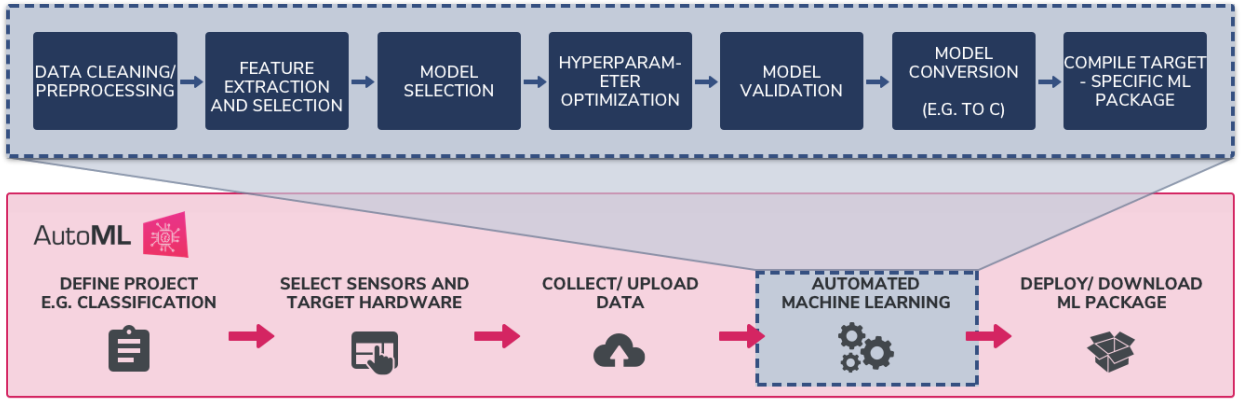
Key Features

- Wide range of ML methods: GBM, XGBoost, Random Forest, Logistic Regression, Decision Tree, SVM, CNN, RNN, CRNN, ANN, Local Outlier Factor, and Isolation Forest
- Easy-to-use interface for labeling, recording, validating, and visualizing time-series sensor data
- On-device inference optimized for low latency, low power consumption, and a small memory footprint
- Supports Arm® Cortex™- M0 to M4 class MCUs
- Automates complex and labor-intensive processes of a typical ML workflow – no coding or ML expertise required!

Target Markets/Applications

- Industrial Predictive Maintenance
- Smart Home
- Wearables
- Automotive
- Mobile
- IoT

QEEXO AUTOML: END-TO-END MACHINE LEARNING PLATFORM



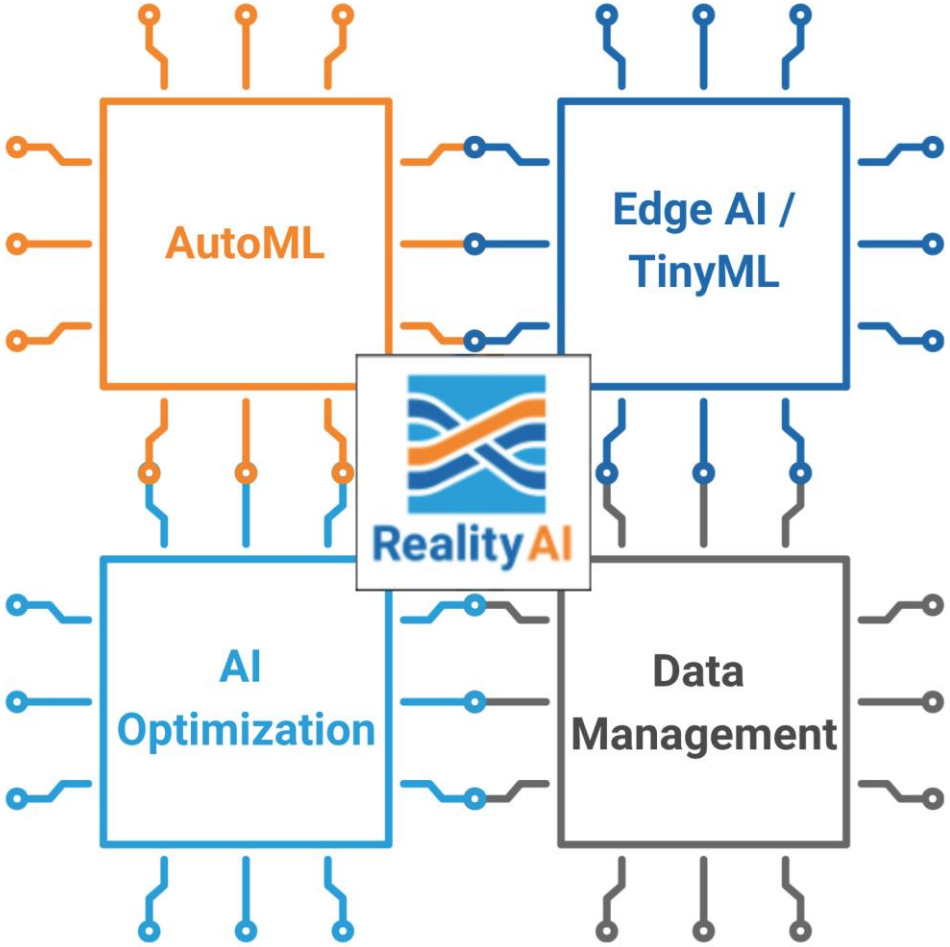
For a limited time, sign up to use Qeexo AutoML at automl.qeexo.com for FREE to bring intelligence to your devices!

Reality AI®

Engineering Solutions for the Edge

Next-Generation AI Tools for Product Development

- Extensive, highly-optimized feature spaces
- Super-compact code for MCUs and Gateways
- Sensor selection and placement analysis
- AI-driven component specs
- Automated data quality checks
- Data collection, augmentation & labeling services
- No open source - clean licensing

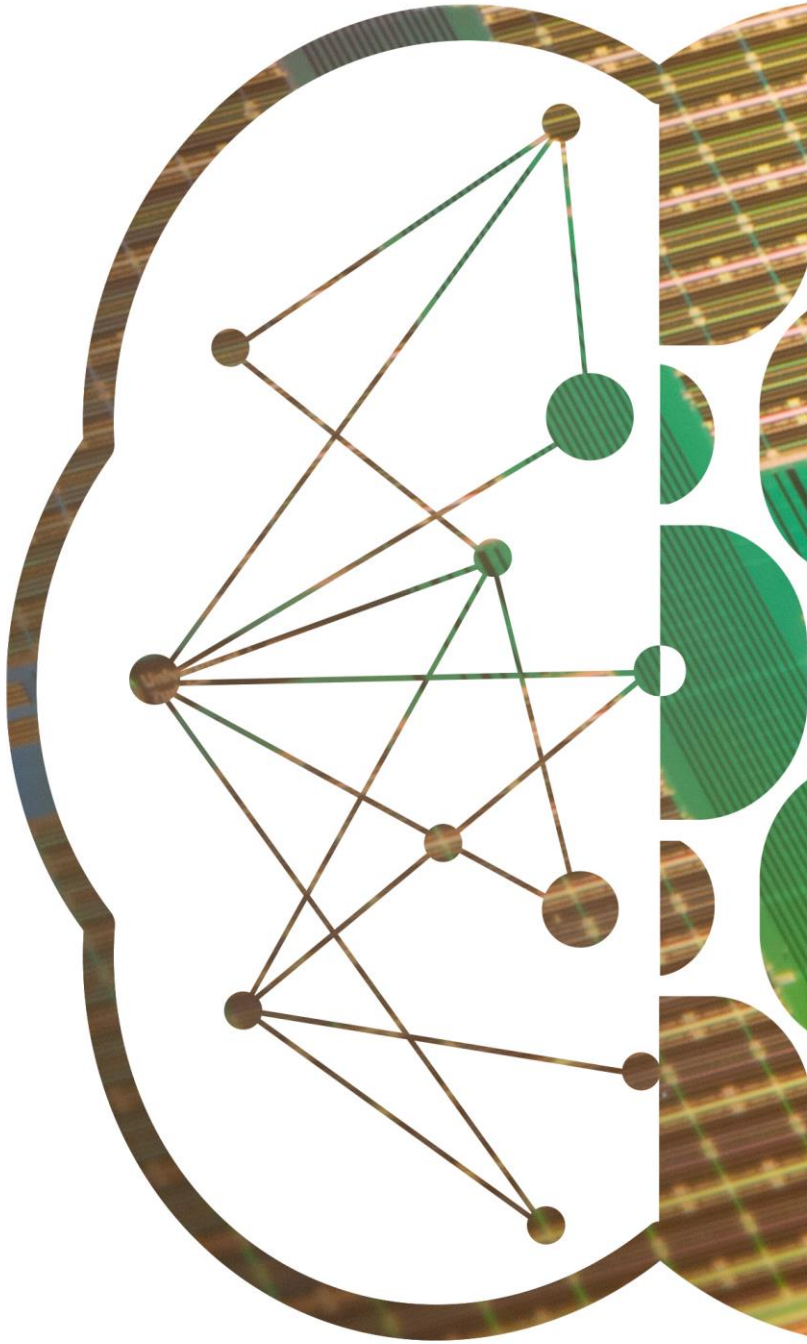


Get started w/ a special tinyML Talks offer for corporate customers: https://reality.ai/get-started

SynSense

SynSense (formerly known as aiCTX) builds **ultra-low-power** (sub-mW) **sensing and inference** hardware for **embedded, mobile and edge** devices. We design systems for **real-time always-on smart sensing**, for audio, vision, bio-signals and more.

https://SynSense.ai



Next tinyML Talks

Date	Presenter	Topic / Title
Tuesday, September 15	**Hiroshi Doyu** Senior researcher, Ericsson Research **Vikrant Tomar** Founder & CTO, Fluent.ai Inc. **Sam Myer** Lead ML Developer, Fluent.ai Inc.	TinyML as-a-Service - Bringing ML inference to the deepest IoT Edge Speech Recognition on low power devices

Webcast start time is 8 am Pacific time

Each presentation is approximately 30 minutes in length

Please contact talks@tinyml.org if you are interested in presenting

TINY ML TALKS webcast

Kristofor Carlson

Kristofor Carlson is a senior research scientist at BrainChip Inc. Previously, he worked as postdoctoral scholar in Jeff Krichmar's cognitive robotics laboratory at UC Irvine where he studied unsupervised learning rules in spiking neural networks (SNNs), the application of evolutionary algorithms to SNNs, and neuromorphic computing. Afterwards, he worked as postdoctoral appointee at Sandia National Laboratories where he applied uncertainty quantification to computational neural models and helped develop neuromorphic systems. In his current role, he is involved in the design and optimization of both machine learning algorithms and hardware architecture of BrainChip’s latest system on a chip, Akida.

brainchip™
Biologically Inspired, Digitally Engineered

The Akida™ Neural Processor: Low Power CNN Inference and Learning at the Edge

Kristofor D. Carlson, PhD
Senior Research Scientist
BrainChip Inc

The Akida Neural System On a Chip (NSoC)

We built an NSoC that performs CNN inference and learning at the edge utilizing neuromorphic design principles

- Neuromorphic design principles:
  - Perform event-based versions of conventional ML algorithms to take advantage of activation sparsity
    - An event is a non-zero activation – Akida only processes non-zero activations
  - Utilize low-bit precision computation with both weights and activations (1 to 4-bit)
  - Co-locate memory and processing
  - Distribute computation across many smaller cores (neural processing units – NPUs) that work in parallel instead of single systolic array like many DLAs
  - Run at lower clock speeds to keep overall power consumption low
  - Implement a proprietary, on-chip, unsupervised learning algorithm

The Akida Neural System On a Chip (NSoC)

We built an NSoC that performs CNN inference and learning at the edge utilizing neuromorphic design principles

- Neuromorphic design principles:
  - **Perform event-based versions of conventional ML algorithms to take advantage of activation sparsity**
    - An event is a non-zero activation – Akida only processes non-zero activations
  - **Utilize low-bit precision computation with both weights and activations (1 to 4-bit)**
  - Co-locate memory and processing
  - Distribute computation across many smaller cores (neural processing units – NPUs) that work in parallel instead of single systolic array like many DLAs
  - Run at lower clock speeds to keep overall power consumption low
  - **Implement a proprietary, on-chip, unsupervised learning algorithm**

The Akida Neural System On a Chip (NSoC)

We built an NSoC that performs CNN inference and learning at the edge utilizing neuromorphic design principles

- Neuromorphic design principles:
  - **Perform event-based versions of conventional ML algorithms to take advantage of activation sparsity**
    - 40-60% base reduction in MACs compared with non-event-based hardware
  - **Utilize low-bit precision computation with both weights and activations (1 to 4-bit)**
    - 50% reduction (or greater) in required memory & bandwidth when compared to 8-bit machine learning accelerators
  - **Implement a proprietary, on-chip, unsupervised learning algorithm**
    - Retraining at the edge instead of retraining in the cloud

Akida Software Development Environment (ADE) and Training Workflow

Akida Software Development Stack

Akida™ Chip Simulator
pip install akida

Training tool (CNN2SNN)
pip install cnn2snn

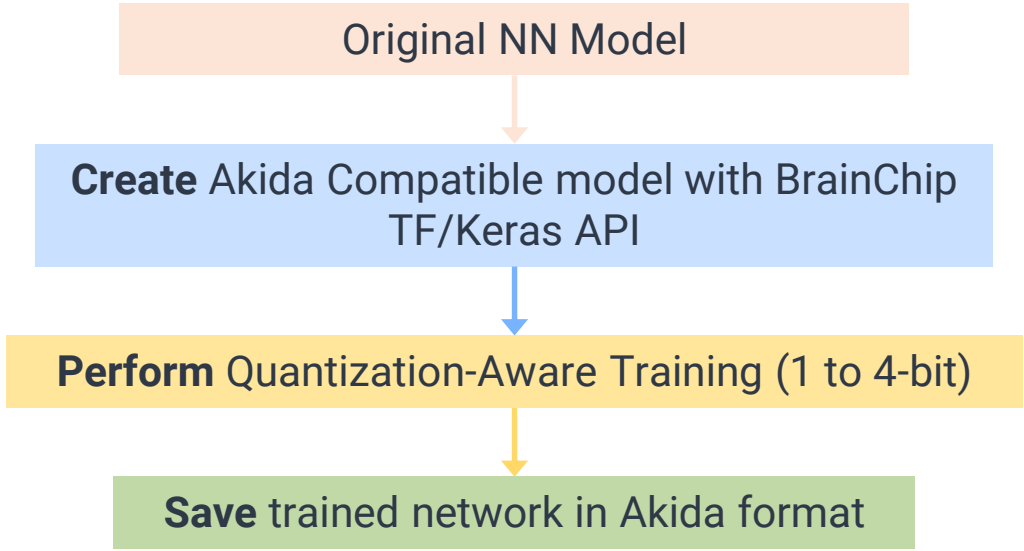
Models
pip install akida-models

python Package Index

TensorFlow

Keras

CNN2SNN Training Tool Workflow



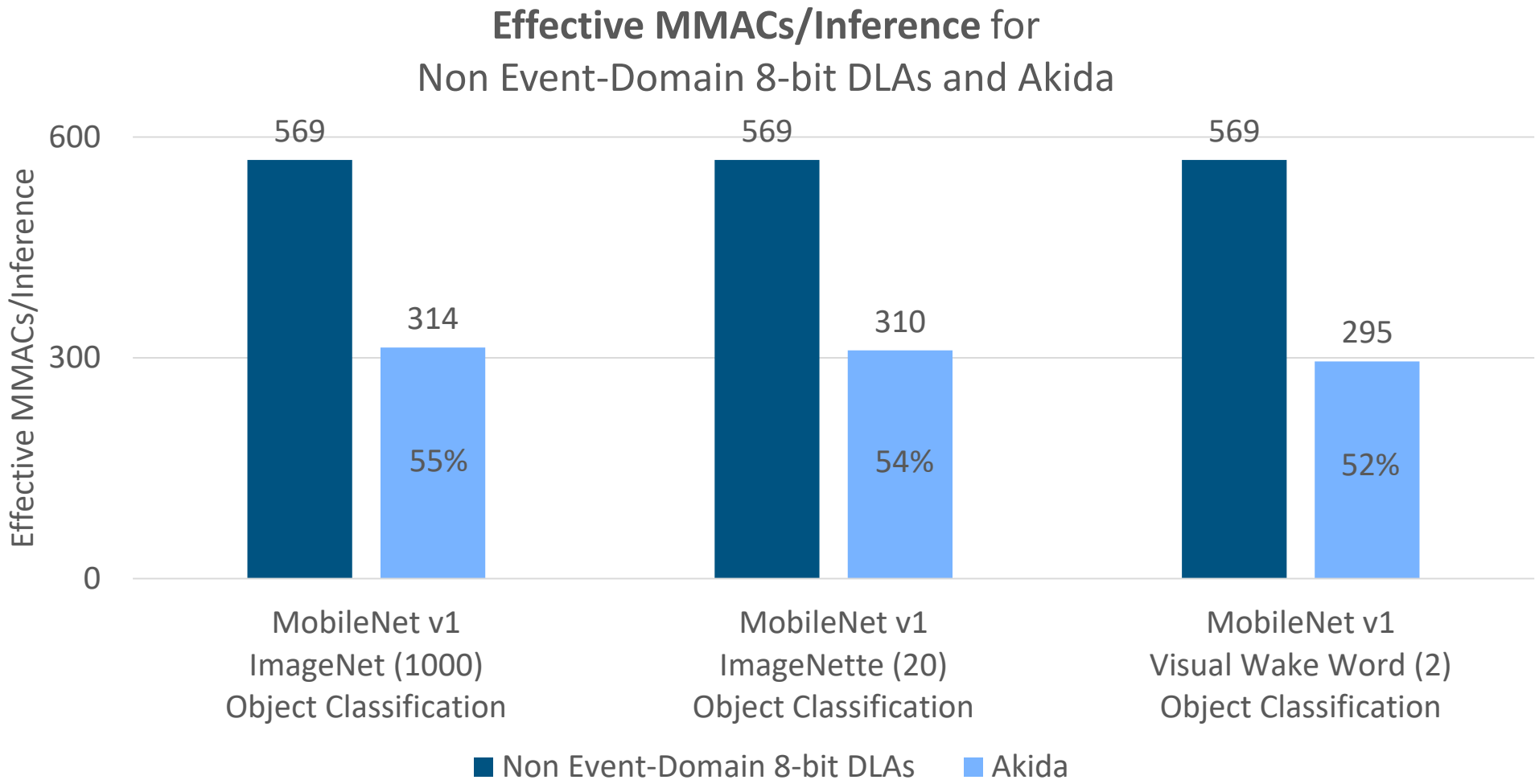
https://doc.brainchipinc.com/

Event-Based Computation

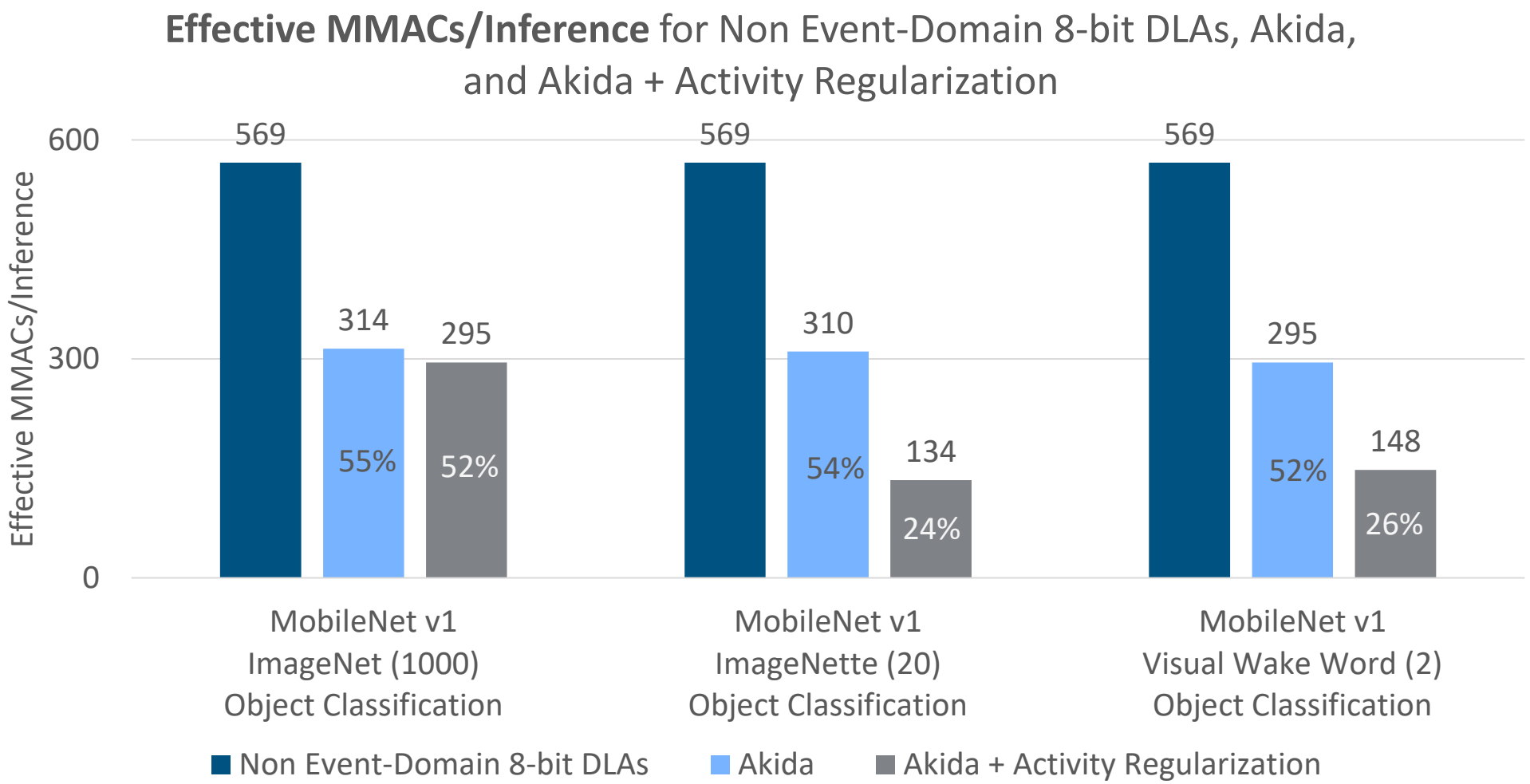
Key Aspects of Event-Based Processing

- Implement event-based versions of convolutions and dot products from the ground-up to minimize overhead
  - **No loss in accuracy** – algorithm is different, but core calculation is identical
  - Higher **activation sparsity** leads to fewer operations
- Batch Normalization gives us 40-60% activation sparsity as a starting point
  - On average, ReLUs are centered around zero and ~50% of the outputs are zero
- We further increase activation sparsity by using **activity regularization** during training
  - **Activity regularization** is the process of adding more information to the loss function to balance the model's accuracy and activity sparsity
  - We used this built-in functionality in TensorFlow while training our models

Akida Utilizes Activation Sparsity to Reduce Computation



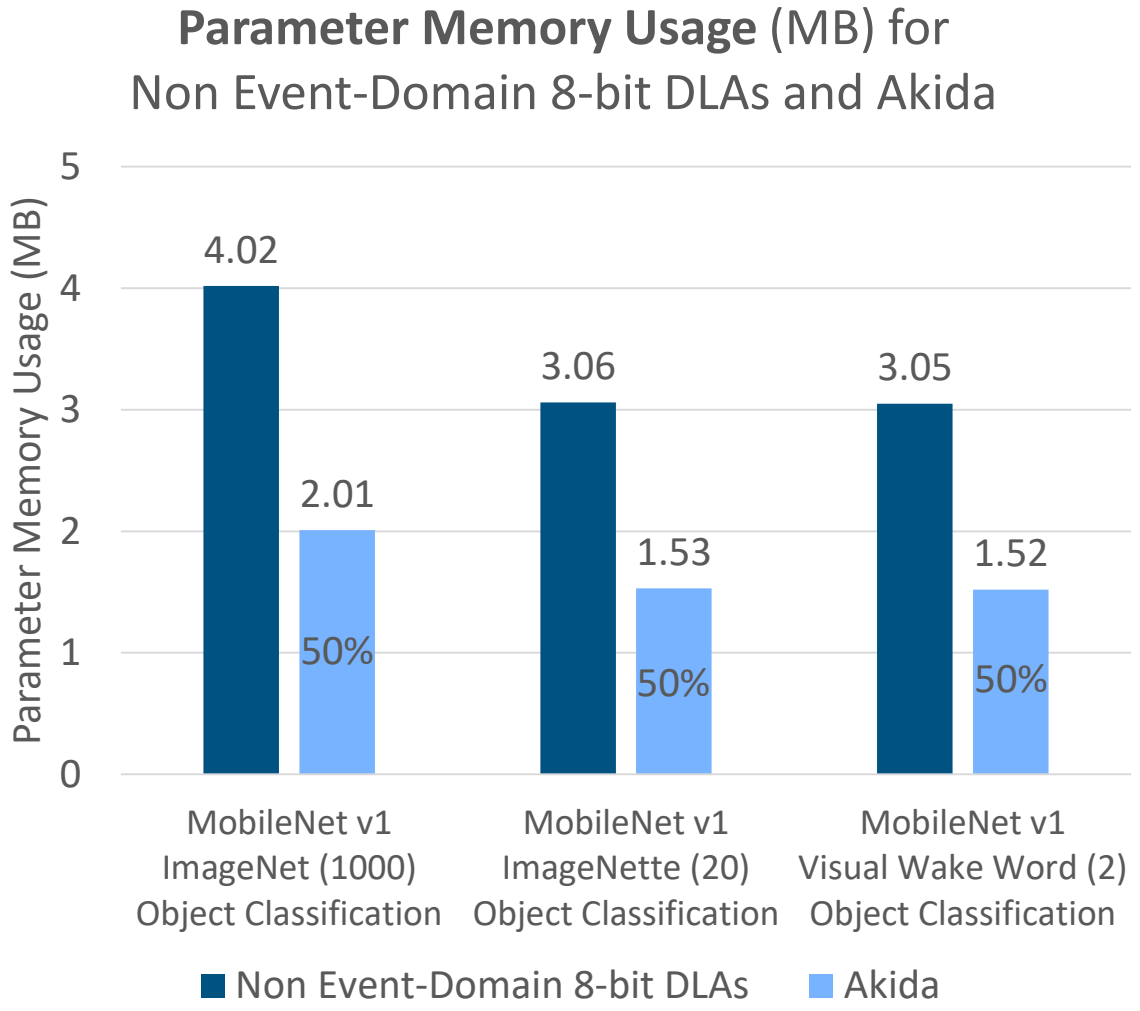
Activity Regularization Increases Activation Sparsity and Further Reduces Computation



Low-Bit Precision Parameters and Activations

Akida Utilizes Low-Bit Precision to Reduce Memory/Bandwidth

- Akida uses 1-4 bits for activations and parameters
  - 50% (or greater) reduction in memory & bandwidth compared to 8-bit hardware
- We currently perform quantization-aware training to preserve accuracy
- Multiple research groups preserve accuracy with post-training 4-bit quantization*



*Banner, R., et al (2019) Advances in NIPS

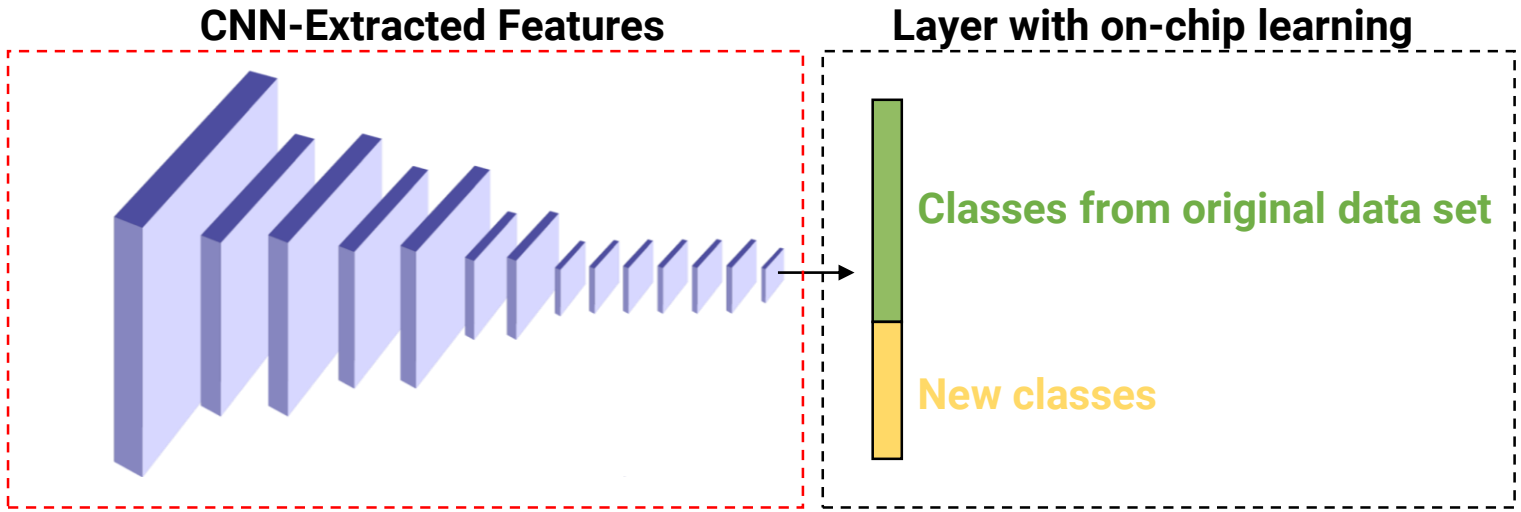
Selected BrainChip Quantization Results

Model	Dataset	# Classes	Weight/Activation Quantization	Quantized Accuracy	32-Bit Float Accuracy
DS-CNN 24K parameters	Google Speech Commands	30	4/4	91.7%	92.0%
MobileNet 224 0.25 200K parameters	Visual Wake Word	2	4/4	89.7%	90.7%
MobileNet V1 2.7M parameters	CIFAR10	10	4/4	93.1%	93.5%
MobileNet V1 4.2M parameters	ImageNet 1000	1000	4/4	68.8%	71.4%
MobileNet SSD 300 5.8M parameters	VOC	20	4/4	65.4%	66.9%
VGG 14.0M parameters	CIFAR10	10	2/2	90.7%	93.2%

Edge Learning

Edge Learning with Akida On-Chip Learning

1. Train CNN feature extractor offline on original dataset
2. Replace last classifier layer with Akida layer capable of on-chip learning
3. Perform few-shot learning: learn from a few samples
   a) original classes (green)
   b) new classes (yellow) – should share similar features with original classes



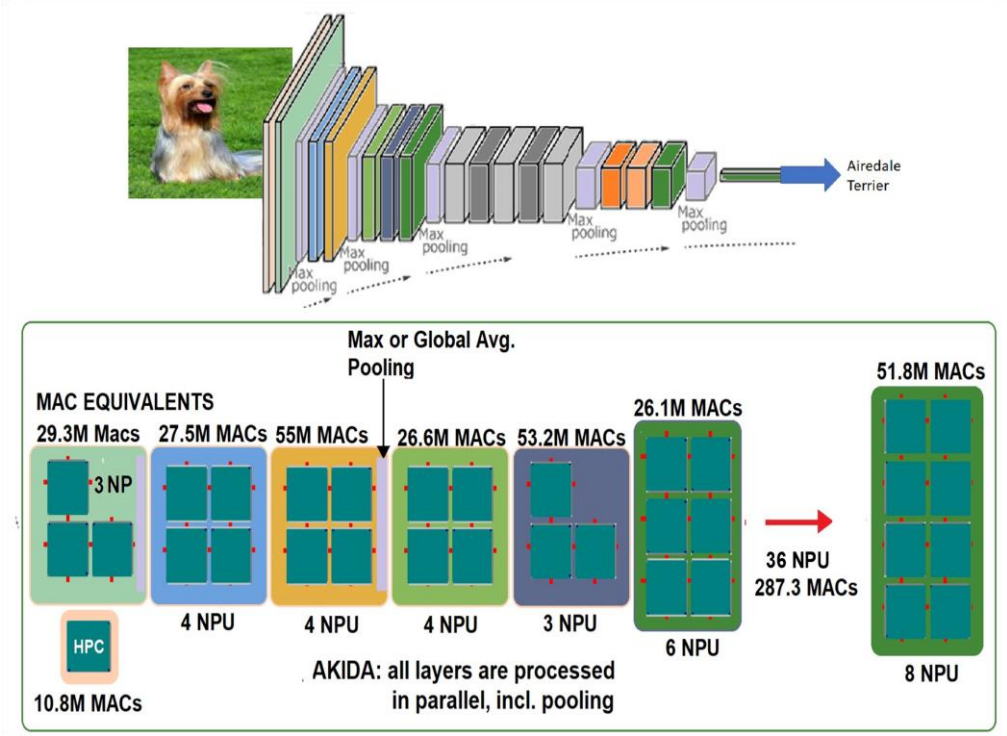
- We have demonstrated edge learning for:
  - Object detection using MobileNet trained on the ImageNet dataset
  - Keyword spotting using DS-CNN trained on the Google Speech Commands dataset
  - Hand gesture classification using small CNN trained on a custom DVS events dataset

Edge Learning with MobileNet and ImageNet

MobileNet V1 ImageNet Classification on Akida Simulator

- MobileNet V1
  - 30 layers with 569 equivalent MMACs per inference
  - Entire network mapped onto separate Akida NPUs (4.2 M parameters) consuming 2 MB memory
  - Ran on Akida Chip simulator

- On-Chip Learning Demo
  - Pretrained on ImageNet
  - Replaced last layer with Akida learning layer
  - Performed 1-shot learning
    - New classes do not need to be part of the original data set



Akida Incremental Edge Learning Demo

https://www.youtube.com/watch?v=BLNr-KXPxhE

AKD1000

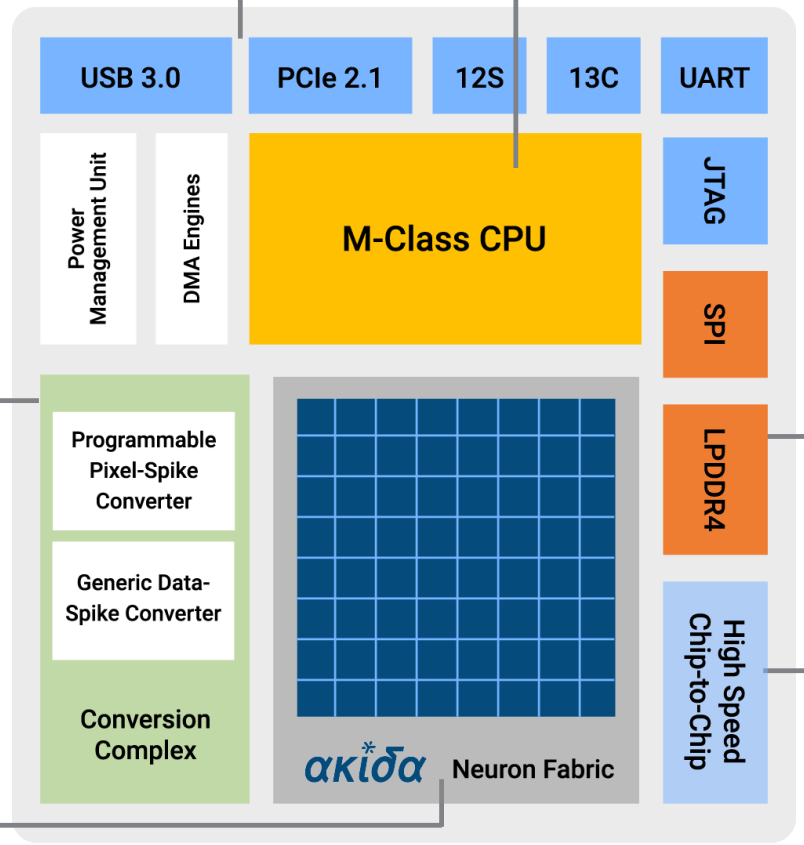
AKD1000 - Complete NSoC AI Edge Solution

Single Platform for CNN inference with on-chip learning

Data input Interfaces
- PCI-Express 2.1 x2 Lane Endpoint
- USB 3.0 Endpoint
- I3S, I2C, UART, JTAG

Conversion Complex
- Pixel-Spike Converter
- SW Data-Spike Encoder
  - Any multivariable digital data
  - Sound, pressure, temp others

Flexible Akida Neuron Fabric
- Implements 80 NPUs
- All Digital logic with SRAM (8MB)
- Also Available as Licensed IP Core
- First Implementation : TSMC 28nm.



On-Chip Processor
- M-Class CPU with FPU & DSP
  - System management
  - Akida Configuration

External Memory Interfaces
- SPI FLASH for boot/storage
- LPDDR4 Program /Weights

Multi-Chip Expansion
- PCIe 2.1 2 lane root complex
- Connects up to 64 devices

Summary

- Compared to 8-bit, non event-based hardware, Akida runs CNN inference with:
  - 50% (or greater) reduction in required memory and bandwidth
  - 40–75% reduction in MACs using event-based design and activity regularization
- Runtime software manages all configuration and network loading
  - Application-level API similar to Tensor-flow/Keras
- Incremental on-chip learning from few samples
- Available as a chip AKD1000 or Embedded IP in your SoC

Questions?

Contact Information:
Anil Mankar – amankar@brainchip.com
Kristofor Carlson – kcarlson@brainchip.com

https://brainchipinc.com/
https://doc.brainchipinc.com/

Copyright Notice

www.tinyML.org