nearbAI, the cutting-edge AI accelerator

nearbAI is easics' flexible embedded AI engine. It performs real-time ultra-low latency pattern recognition using artificial neural network(s), close to your sensor(s). It efficiently runs your Deep Neural Network (DNN) inference: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Temporal Convolution Networks (TCN). This hardware component integrates readily right next to your sensor(s). It outputs structured data with the lowest possible inference latency and power consumption.

nearbAI for embedded AI can be deployed on any FPGA System-on-Chip (SoC) device. The DNN layers are hardware-accelerated using programmable logic. An embedded micro-controller (MCU) inside the FPGA runs all embedded software, including any pre- and post- processing.

With *nearbAI*, easics offers an optimized accelerator for the DNNs in today's and tomorrow's applications. *nearbAI* is highly parameterizable to match your requirements efficiently. It comes with an extensive software development suite, which is used to map your neural network on the *nearbAI* demonstrator. The provided API is easy to use and offers a fast and efficient bridge to your software application.

Xilinx® Zynq UltraScale+™ MPSoC ZCU104 Evaluation Kit

why FPGA as an AI accelerator

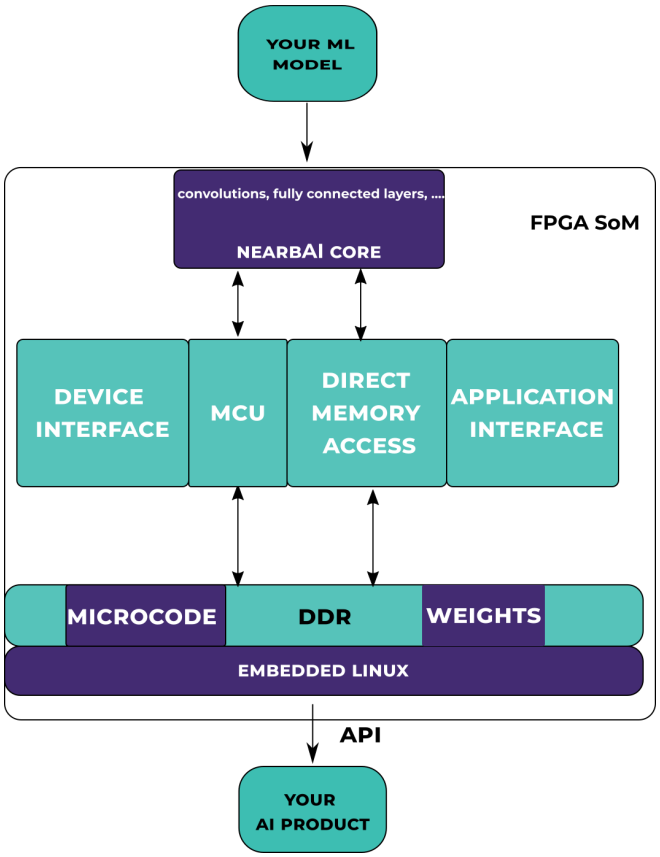
An FPGA is a blank canvas of logic resources suitable for real-time applications with AI. It offers the following benefits:

- ultra-low latency
- better performance-per-Watt than GPUs
- scalable: the same *nearbAI* instance can be used on both low-end & high-end FPGAs
- reprogrammable in the field to support future DNN architectures
- interfaces tailored to your needs
- fixed-point weights and tensors for a small memory footprint, while maintaining the desired accuracy
- medical, industrial, automotive and space grade devices available
- operating temperature: -40 … +85 ℃
- supports product life cycles of 10+ years

nearbAI demonstrator and evaluation kit

The ZCU104 board containing a Xilinx® Zynq UltraScale+™ or the Achilles board containing an Intel® Arria® 10 SoC are the right environment for your *nearbAI* evaluation. The evaluation kit contains the following items:

- ZCU104 or Achilles hardware
- *nearbAI* evaluation core (top-level bitmap)
- we compile **your neural network** including microcode and quantized weights
- API to load your weights, send real-time data, and get the inference results. This API is available in Python or in C++.
- Linux kernel and drivers on SD card
- standard interfaces:
  - Ethernet
  - other interfaces on demand
- user guide



Instant DNN to demonstrator conversion

nearbAI software development suite

The software suite supports neural networks in the ONNX format:

- *Estimator*: estimates hardware resource utilization and inference time for the DNN model(s), parameters and FPGA device(s) of your choice.
- *Network Compiler*:
  - performs quantization
  - generates microcode that contains the sequence and memory allocation
- *Core Generator*: generates a parameterized VHDL-core, simulation and synthesis scripts.

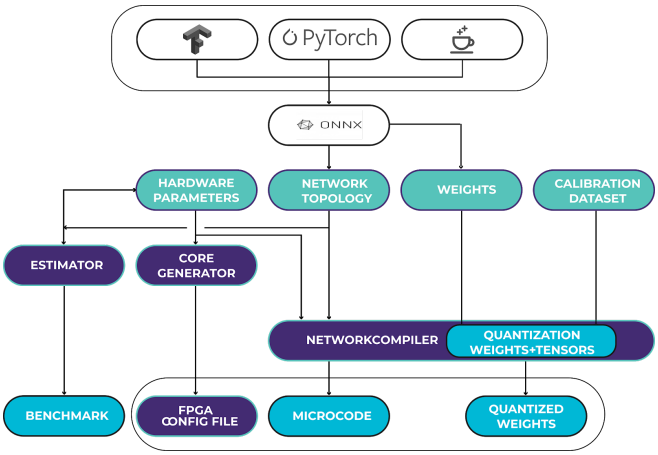
nearbAI baseline core

A baseline *nearbAI* IP core natively supports a rich set of mathematical operations (kernels), it supports these DNNs out-of-the- box, besides many more:

- CNN: ResNet, YOLO, MobileNet
- RNN: DeepSpeech

easics' baseline *nearbAI* core supports these operations, being the primitives (kernels) of the vast majority of standard DNNs:

- 2D convolution
- 2D ConvLSTM layers
- depthwise convolutions
- dilated convolutions
- transposed convolutions
- matrix multiplications
- fully-connected layers
- bias
- max pooling, average pooling
- ReLU, ReLU6, Leaky ReLU
- easics will add primitives on your request, both standard and proprietary ones.



nearbAI software development suite

parameterization of nearbAI

The *nearbAI* core is optimized taking into account your application-specific needs:

- your standard or custom DNN model(s)
- latency, throughput (inference rate),power consumption, and cost (hardware resources)

- DNN accuracy: 8 or 16 bit data and weights

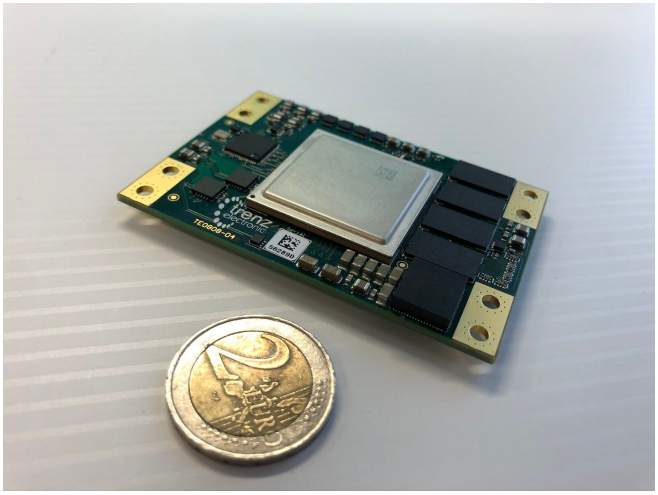
The optimization is done by tuning these parameters:

- input & output channels or the number of MAC of the convolution engine
- clock frequency
- internal memory buffers and external memory bus
- word size of the tensors and the weights

nearbAI production deployment

- Intel® Arria® 10
- Intel® Cyclone® 10
- Xilinx® Ultrascale+™
- FPGA System-on-Module (SoM)
- Other FPGAs on demand

- Multiple sensors connected to the same FPGA in a stackable design dedicated for your application.
- ASIC



FPGA System-on-Module:
Trenz - TE0XX Zynq UltraScale+ MPSoC

Get a *nearbAI* engine tailored precisely to your needs:

- bring your neural network and create your embedded AI demonstrator
- the *nearbAI* API is easy to use and offers a fast and efficient bridge to your software application
- customize your *nearbAI* engine according to your DNN model(s), performance requirements, latency and power consumption target.
- let the *Estimator* tool assist you with making trade-offs and guiding you to the hardware platform that is best suited for your *nearbAI* deployment https://nearbai.easics.com
- benefit from *nearbAI*'s low hardware cost, thanks to its record-breaking MAC efficiency, usually north of 95%
- *nearbAI* supports CNNs, RNNs and TCNs on the same instance, resulting in unparallelled flexibility
- save person-months worth of effort on porting your software implementation to FPGA, by relying on the *nearbAI* software development suite
- perform early optimizations on real-time FPGA prototype hardware
- entrust easics' *nearbAI* support team with on-demand integration support of the *nearbAI* engine in your application
- ASIC developers use *nearbAI* on FPGA as an emulation platform for the design and verification of their custom AI accelerators targeting ASIC. This way, the software development starts simultaneously with the hardware development. *nearbAI* offers a cost efficient path to deploy AI on ASIC.