Energy-efficient On-device Processing for Next-generation Endpoint ML

Tomas Edsö
Senior Principal Engineer, Arm Machine Learning Group
Arm Enables AI Everywhere, On Any Device

Arm’s AI platform delivers comprehensive hardware IP, software frameworks, and ecosystem.

AI-enabled IoT device shipments forecast to increase by almost 20% per year through 2024*

*Source: Arm and industry data
# Best-in-class Solution Optimized for Endpoint AI

<table>
<thead>
<tr>
<th>Performance</th>
<th>Optimization</th>
<th>Accelerated Design and Development</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M55</td>
<td>Arm Custom Instructions*** and configuration options</td>
<td>Corstone-300 reference design for faster and more secure system-on-chip development</td>
</tr>
<tr>
<td>Versatile ML performance: Up to 15x ML uplift*</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Ethos-U55</td>
<td>Configurable 32-256 MACs</td>
<td></td>
</tr>
<tr>
<td>First microNPU for Cortex-M</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Dedicated ML performance: Additional 32x ML uplift**</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*Compared to previous Cortex-M generations

**Compared to the Cortex-M55

***available in 2021
Unified Software Development: Fastest Path to Endpoint AI

Multiple software development flows
- Harder to program and debug
- More complex, longer time to market

Unified software development flow
- Works with common ML frameworks and existing tools
- More productivity, faster time to market
Cortex-M55 Processor

Arm’s most AI-capable Cortex-M processor and the first to feature Arm Helium vector processing technology.
Cortex-M55: The most AI-capable Cortex-M CPU

- First CPU based on Arm Helium technology
  - Energy-efficient and configurable with vector processing capabilities
  - Delivers up to 5x DSP performance and up to 15x ML performance*
  - Versatile capability for both classical ML and NN inference
- Advanced memory interfaces for fast access to ML data and weights
- Arm TrustZone security, accelerating the route to PSA Certified

*Compared to previous Cortex-M generations
Simplified Software Development Based on a Unified Programmer’s View

Unified toolchain for control code, signal processing and ML

- **CMSIS-DSP library**
  - Optimized library for signal processing and classic ML techniques

- **CMSIS-NN library**
  - Armv8.1-M and Helium
  - Optimized library for ML

- **Armv8.1-M and Helium**
  - Signal processing
  - Machine learning
  - Efficient control code execution
Cortex-M55 and CMSIS-NN performance results

- Quarterly releases of CMSIS-NN
- Continuously increasing performance
- These numbers show current improvements
Ethos-U55 Processor

The first Arm microNPU for Cortex-M based systems
Ethos-U55 overview

- Works alongside Cortex-M55, Cortex-M7, Cortex-M33 and Cortex-M4 processors
- Works alongside on-chip SRAM and system flash
- Accelerates CNN and RNN operators.
- Efficient weight compression
- 8- or 16-bit activations
  Weights are always 8-bit
- 32, 64, 128 or 256 MAC/cc configurations
Typical Ethos-U55 data flow

0. An offline compiled command stream with corresponding compressed weights are put into system Flash.
1. Input activations are put into system SRAM.
2. The host starts Ethos-U55 by defining all memory regions to be used. In particular the location of the command stream and input activations.
3. Ethos-U55 autonomously runs all commands, using SRAM as a scratch buffer. Final results are written to a defined SRAM buffer.
4. Interrupt on completion of writing the final result.
Ethos-U55 interfaces

- 32-bit APB slave for registers access
- Two AXI master interfaces
  - M0: Full read+write AXI master to SRAM
  - M1: Read only AXI master to flash
- Q-channel for clock control
- Q-channel for power control
- IRQ for signaling to host
Mapping of NNs to Hardware using TensorFlow Lite
Offline Optimizer

- Reads a tflite file and identifies subgraphs
- Optimizes scheduling of subgraphs
- Loss-less compression of weights
- Generates commands for microNPU
- Writes out a modified tflite file

Up to 90% SRAM size reduction
Up to 70% model size reduction

Enabling networks not before feasible in embedded systems
Weight Compression

- Neural network weights are a big strain on flash capacity
  - Compression allows larger networks on a device

- Fully connected and RNN layers typically weight bandwidth bound
  - Compression speeds up execution

- Ethos-U55 uses lossless weight compression
  - Operates of quantized model
  - No precision is lost as part of the offline optimization

- Good compression for unmodified weights
  - Normally ~30% reduction of model size

- Great compression if networks have been trained towards sparsity/clustering
  - Can get up to ~80% reduction of model size with insignificant accuracy loss
Network support in Ethos-U55

- Ethos-U55 can completely execute networks that map to the supported operator set
  - For example:
    - Deepspeech_v1
    - RNNoise
    - Wav2letter

- Any unsupported operation fallback to the Cortex-M processor
  - These are accelerated through CMSIS-NN library
  - For most popular networks ‘Softmax’ is the only unsupported operator
  - For example:
    - DSCNN_L
    - MobileNet_v1
    - MobileNet_v2
Ethos-U55 performance results
Using 128 MACs/Cycle configuration of Ethos-U55

Based on early estimates
Smart Speaker Use Case

The combined uplift of Cortex-M55 and Ethos-U55
An example smart speaker pipeline

- The pipeline is a mix of NN and classic signal processing
- Use Cortex-M55 for the classic signal processing
  - With the free optimized signal processing libraries
- Use Ethos-U55 to enable large networks not possible in CPU, such as ASR and NLP
  - With the free optimizer, models fit on realistic embedded SRAM and flash systems
- Use Cortex-M55 along with Ethos-U55 to follow the moving front
  - If a future NN beats classic processing, Ethos-U55 can offload Cortex-M55
  - If a future NN improves using a future, non-supported operator, Cortex-M55 can offload Ethos-U55
Throughput – smart speaker use case

Based on early estimates

70% reduction

On Cortex-M
Throughput – smart speaker use case

Based on early estimates

Heavy workload for Cortex-M

Not to scale
Throughput – smart speaker use case

Based on early estimates

Not to scale

85% reduction

On Cortex-M

On Ethos-U55

© 2020 Arm Limited (or its affiliates)
Example: Typical ML Workload for a Voice Assistant

Latency and energy spent for all tasks listed combined: voice activity detection, noise cancellation, two-mic beamforming, echo cancellation, equalizing, mixing, keyword spotting, OPUS decode, and automatic speech recognition.

- Faster responses
- Smaller form-factors
- Improved accuracy

Latency and energy spent for all tasks listed combined: voice activity detection, noise cancellation, two-mic beamforming, echo cancellation, equalizing, mixing, keyword spotting, OPUS decode, and automatic speech recognition.
Summary
Industry-wide Effort: The Most Extensive AI Ecosystem

Significant silicon partner collaboration

Algorithm, software, tools and RTOS partners

...and others
Summary: Bringing the Benefits of AI to Billions More - Devices

- Unprecedented performance
- Simple software development
- Industry-leading ecosystem

The smallest devices in the world will now participate in and contribute to the AI revolution

Learn more at www.arm.com/AI-endpoint
The Arm trademarks featured in this presentation are registered trademarks or trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

www.arm.com/company/policies/trademarks
Copyright Notice

The presentation(s) in this publication comprise the proceedings of tinyML® Summit 2020. The content reflects the opinion of the authors and their respective companies. This version of the presentation may differ from the version that was presented at the tinyML Summit. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyML.org