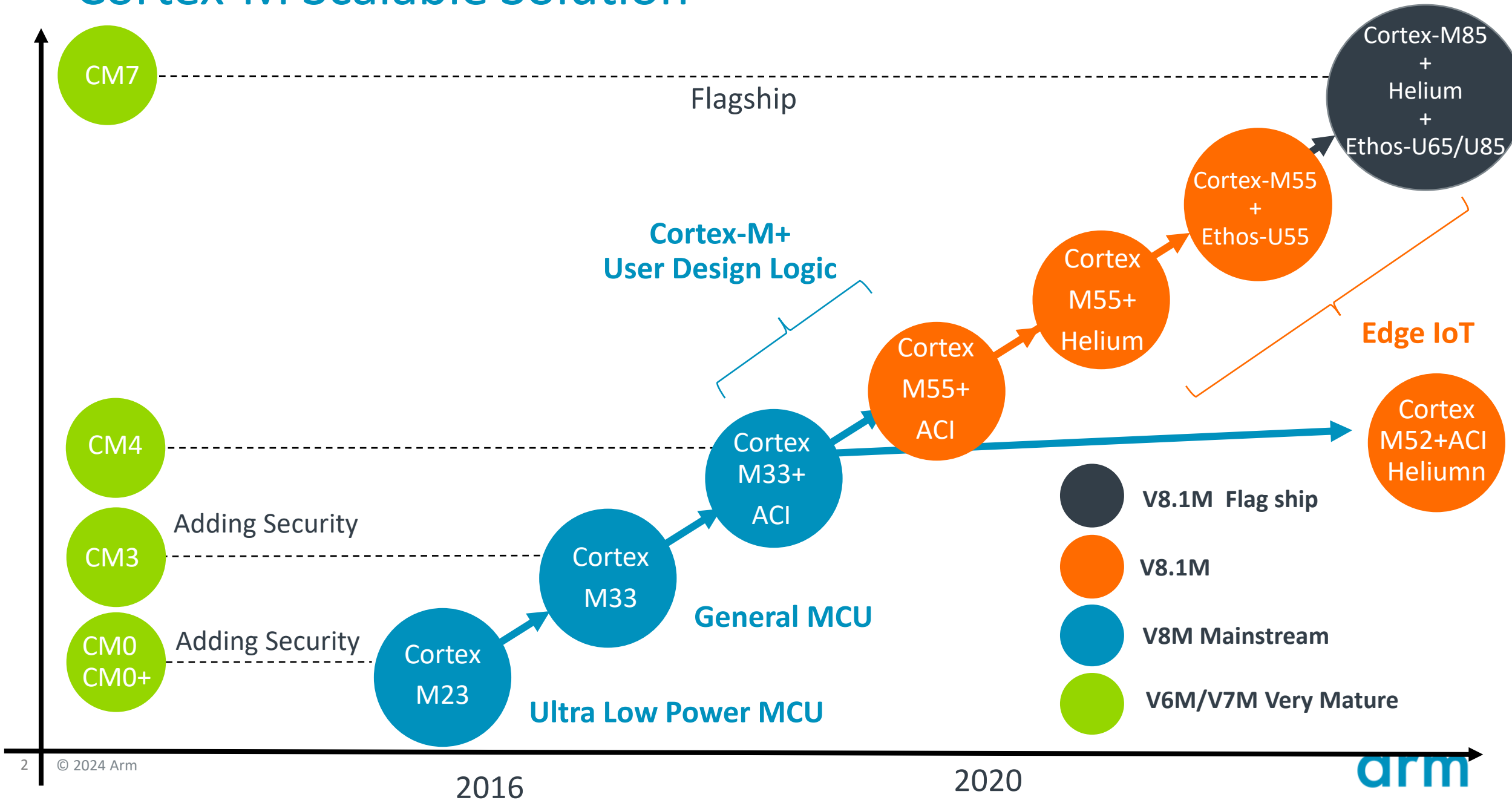
Arm µNPU Ethos-U55 ML inference solution for area and power-constrained device.

Arm K.K. Takuya Mizukami

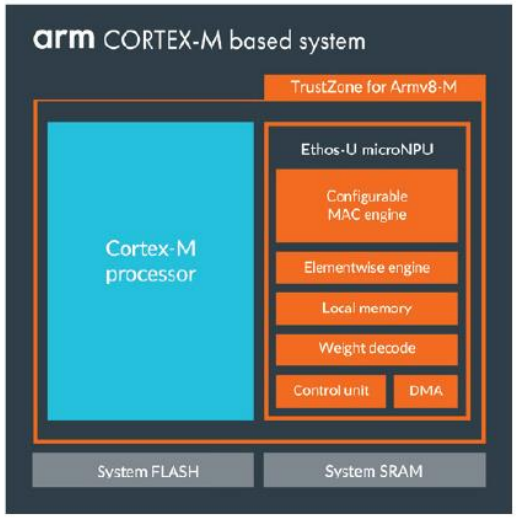
Nov 2024

Cortex-M Scalable Solution



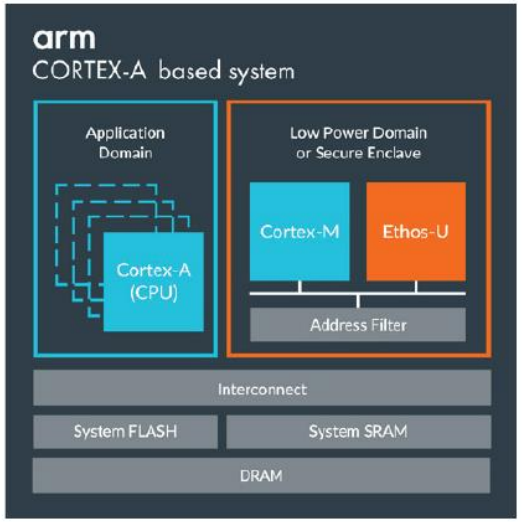
Ethos-U Scalable Solution

Endpoint AI
Cortex-M+Ethos-U55



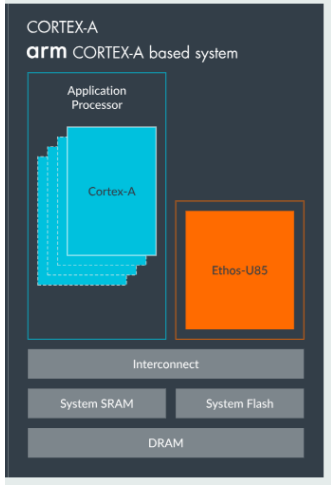
- 64 to 512 GOP/s (at 1GHz)
- 32/64/128/256 MACs/cycle
- Two 64-bit AXI
- SRAM and Flash
- Corstone-300 (Cortex-M55+Ethos-U55)
- Corstone-310 (Cortex-M85+Ethos-U55)

Endpoint AI
Cortex-M+EthosU-65
+
Heterogeneous System
Cprtex-A+(Cortex-M+EthosU65)



- 512GOP/s to 1TOP/s (at 1GHz)
- 256,512MACs/cycle
- Two 128bit AXI
- SRAM, DRAM and/or FLASH
- Corstone-315
(Cortex-M85+Ethos-U65)

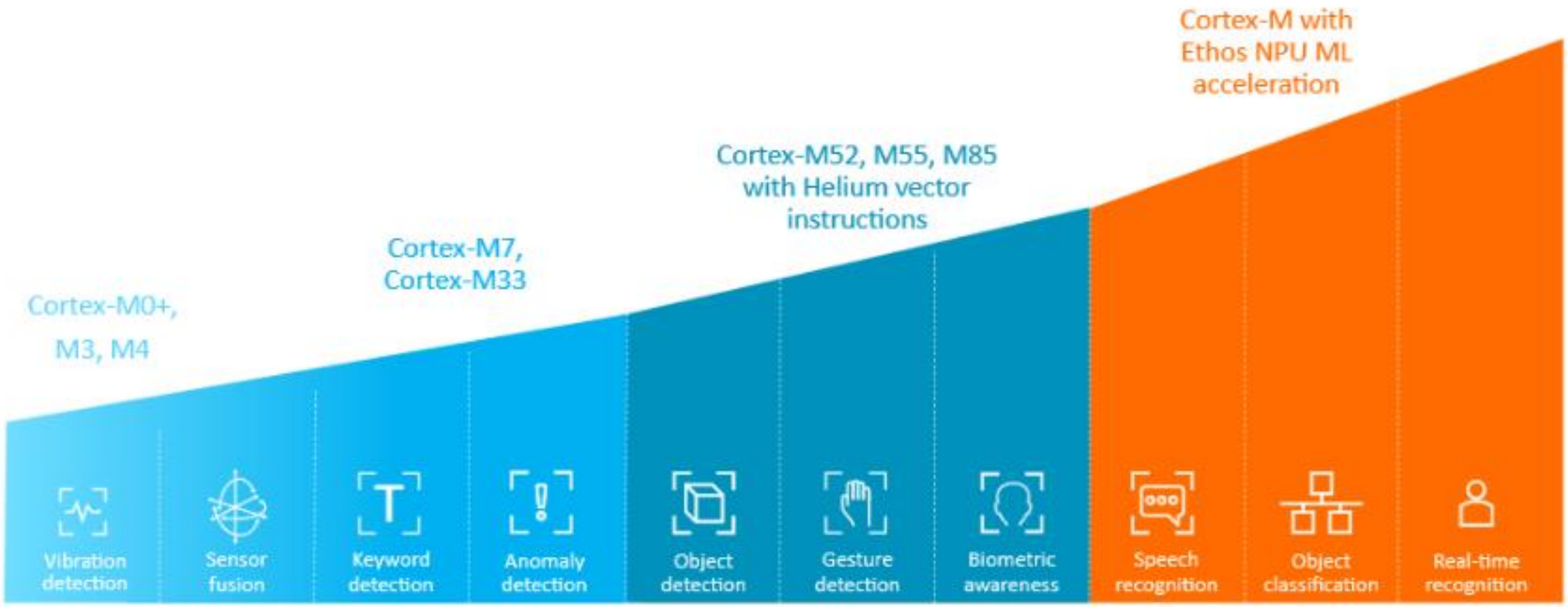
Endpoint AI
Cortex-M+EthosU-85
+
Heterogeneous System
Cprtex-A+(Cortex-M+EthosU85)
+
Cortex-A System
Cortex-A+Ethos-U85



- 256GOP/s to 4TOP/s (at 1GHz)
- 128,256,512,1024,2048MACs/cycle
- Up to six 128-bit AXI5
- SRAM, DRAM and/or FLASH
- Transformer Network support
- Corstone-320
(Cortex-M85+Ethos-U85)

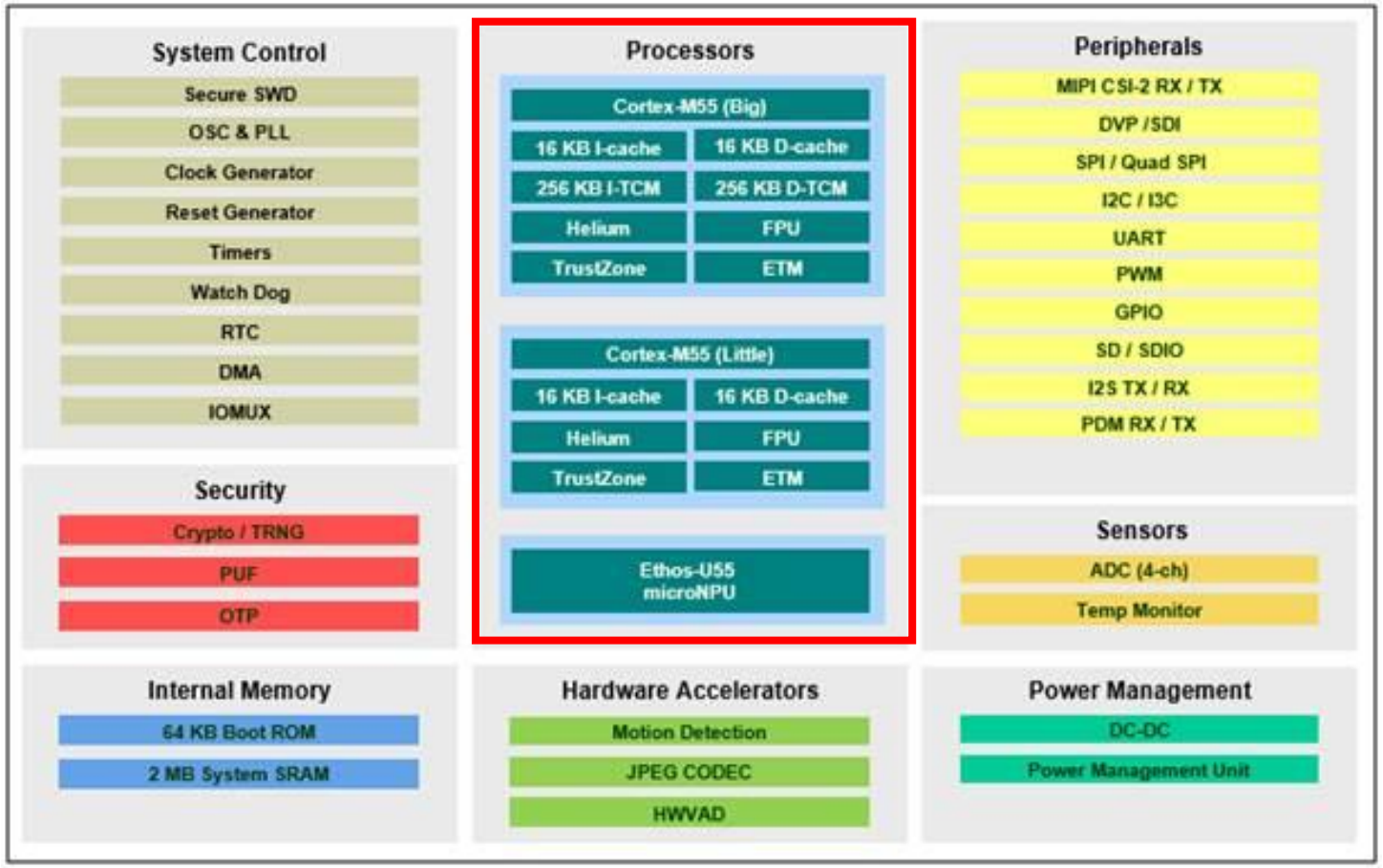
2020 2024

ML edge device use case



ML edge device Typical use cases: Cortex-M and Ethos NPU

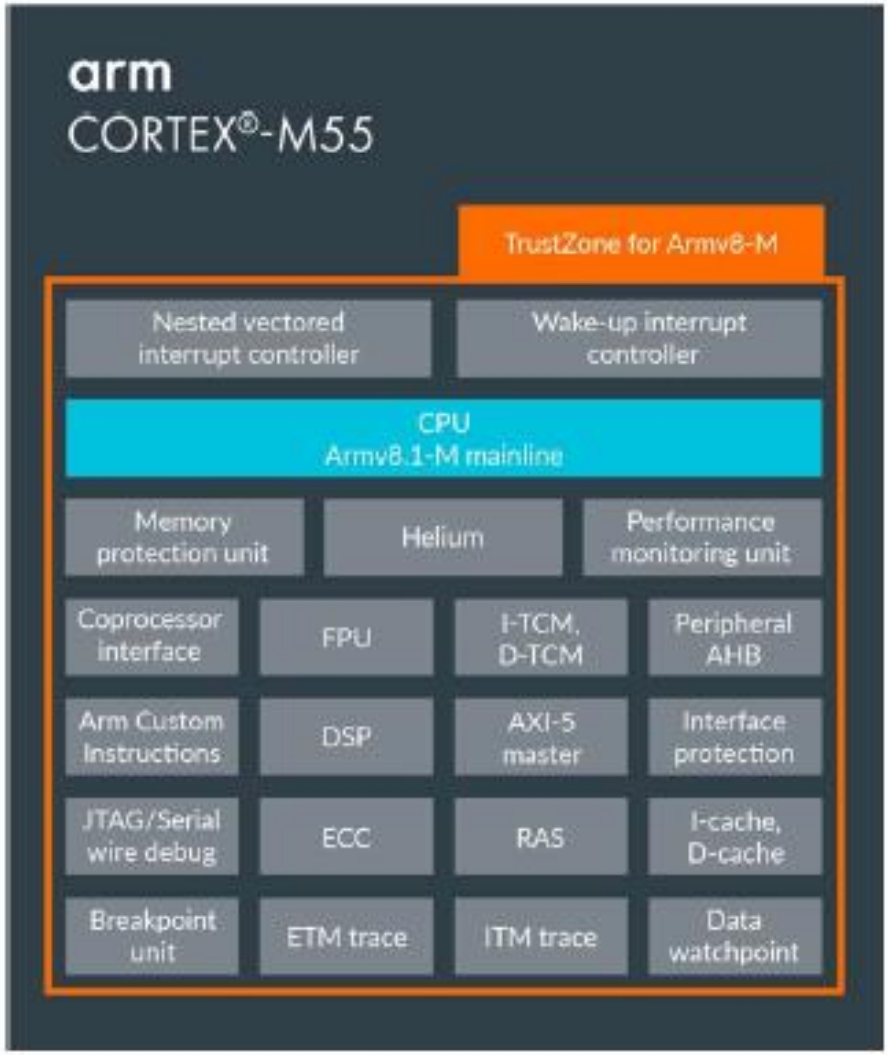
Himax WiseEye2



WiseEye 2 AI Processor block diagram

Refer to https://himaxwiseeyeplus.github.io/

Cortex-M55



- Vector processing
  - 8 vector registers, 128-bit wide, reuse FPU registers
  - Over 150 new instructions (>130 vector instructions)
- Versatile processing capabilities
  - Vectored Integer / Fixed-point : 32-bit, 16-bit, 8-bit
  - Vectored Floating-point : Single precision, half precision arithmetic
  - Scalar Floating-point : Double, single & half precision arithmetic
- Highly configurable design
- Optimized memory system design
- TrustZone Security Extension

arm

Memory System Design

Requirements

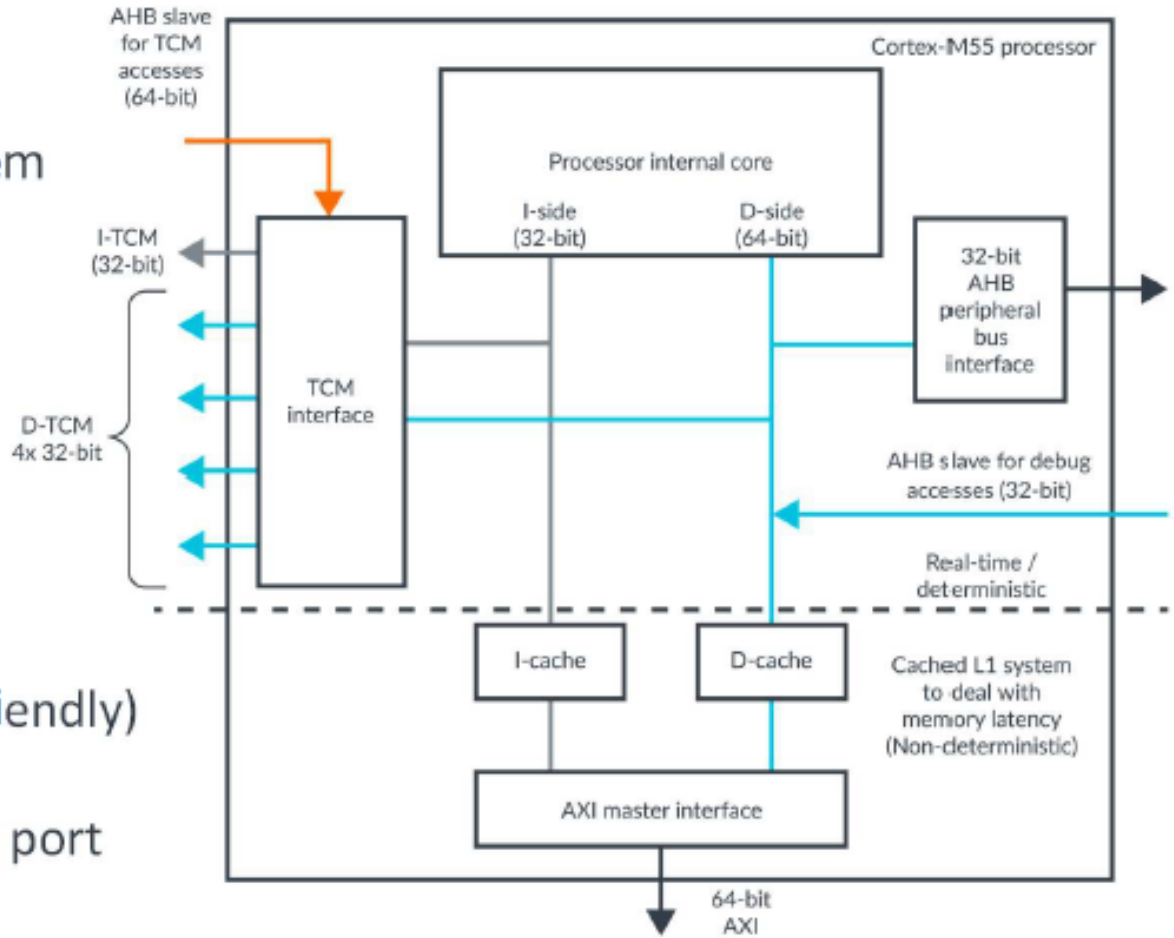
- Real-time, low latency - TCMs
- General purpose - caches

} A two-level memory system

- 64-bit data read/write bandwidth
- 32-bit instruction fetch bandwidth
- Up to two separated data R/W (scatter gather)
- 64-bit bandwidth for DMA accesses to TCM

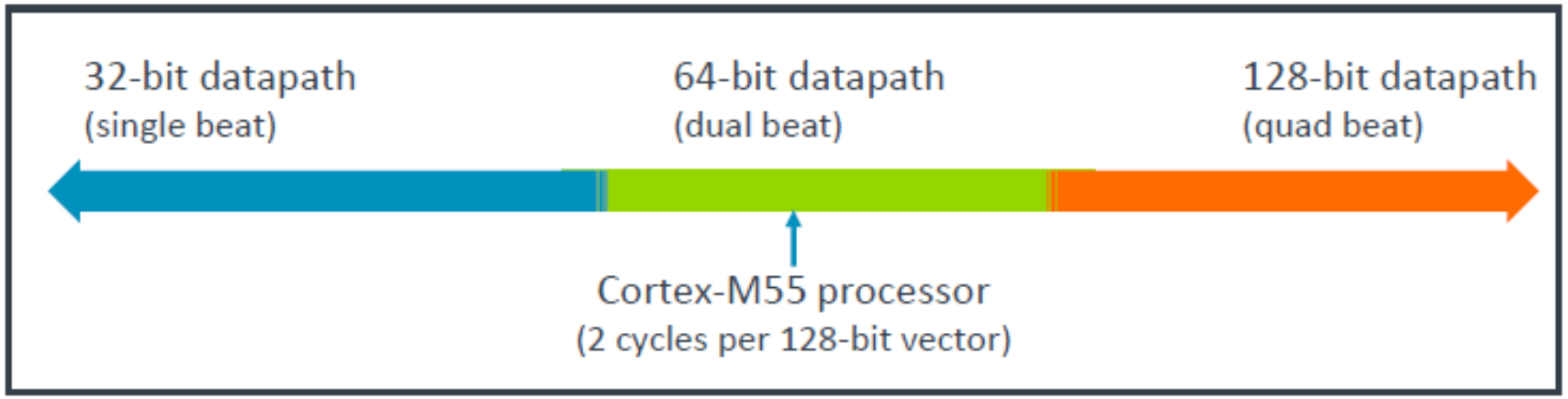
No dedicated DSP memory ports

- TCMs are part of system memory map (C/C++ friendly)
- Up to 16MB I-TCM and 16MB D-TCM
- DMA controller can access to TCM via AHB slave port

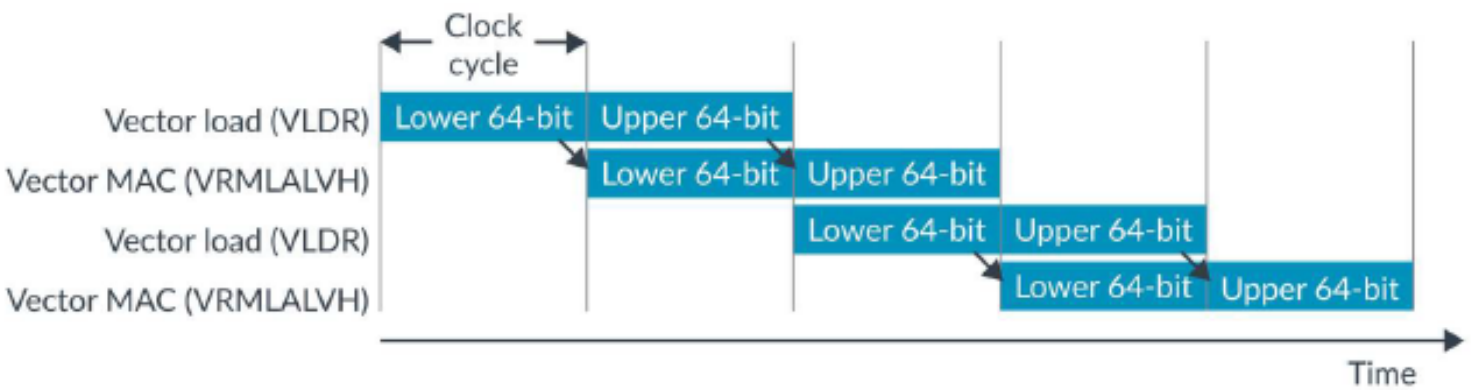


Processor's Vector Pipeline and Datapath

- A balance between performance and power
  - Double ALU area, >4x performance
  - Suitable for short pipeline



- Overlapping instruction execution to enable higher processing efficiency



Cortex-M55 supports

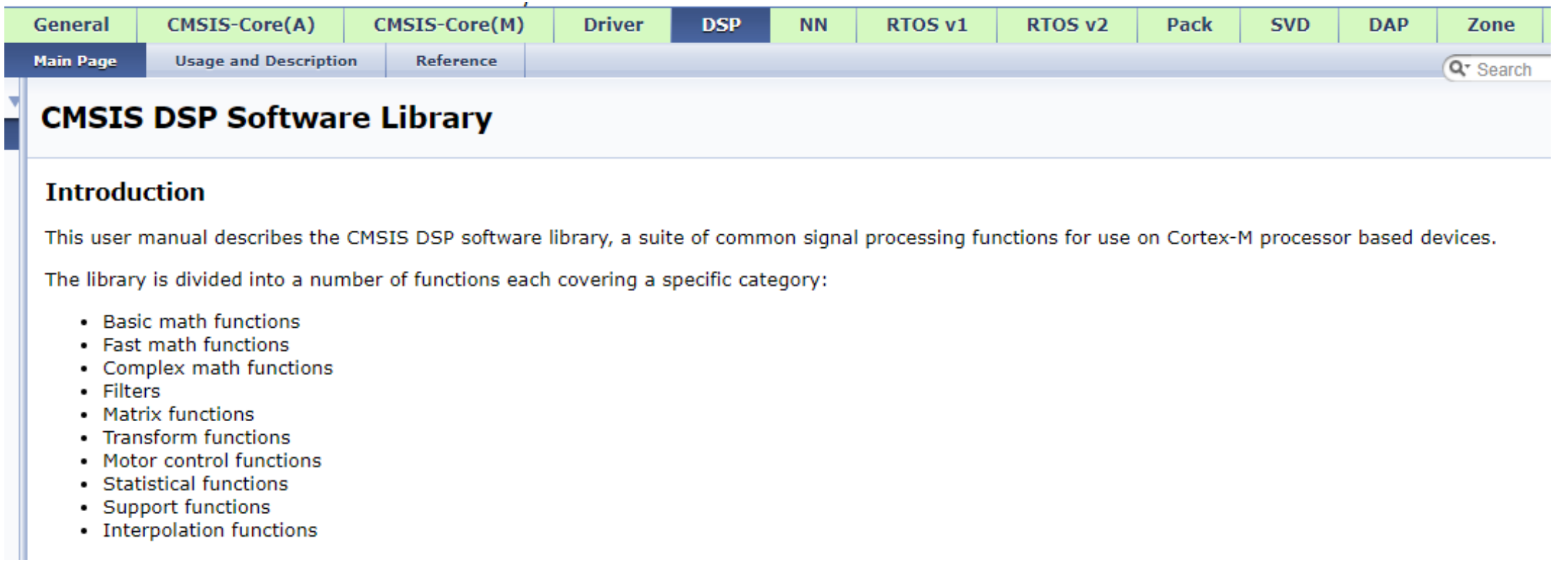
2x Q31/int32
4x Q15/int16
8x Q7/int8
2x fp32
4x fp16

Per clock cycle

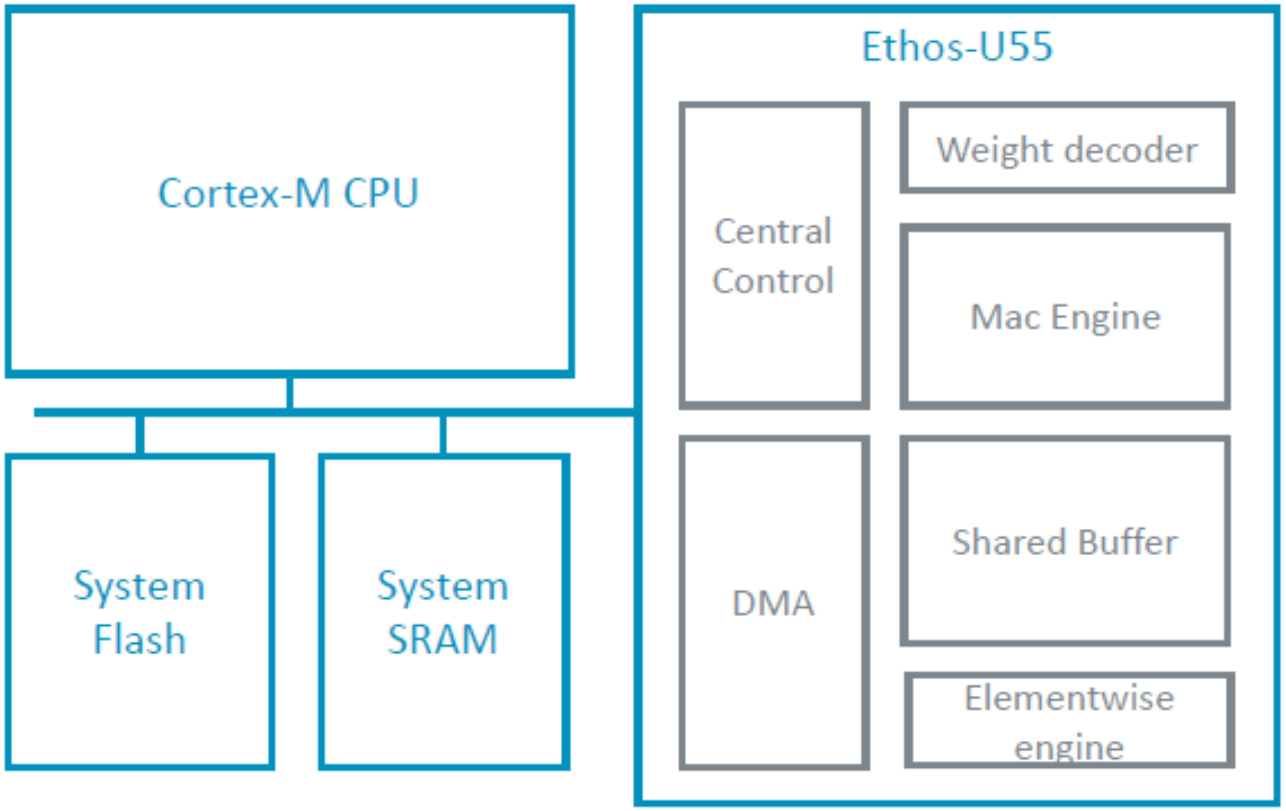
arm

CMSIS-DSP and CMSIS-NN

- https://arm-software.github.io/CMSIS_5/DSP/html/index.html



Ethos-U55

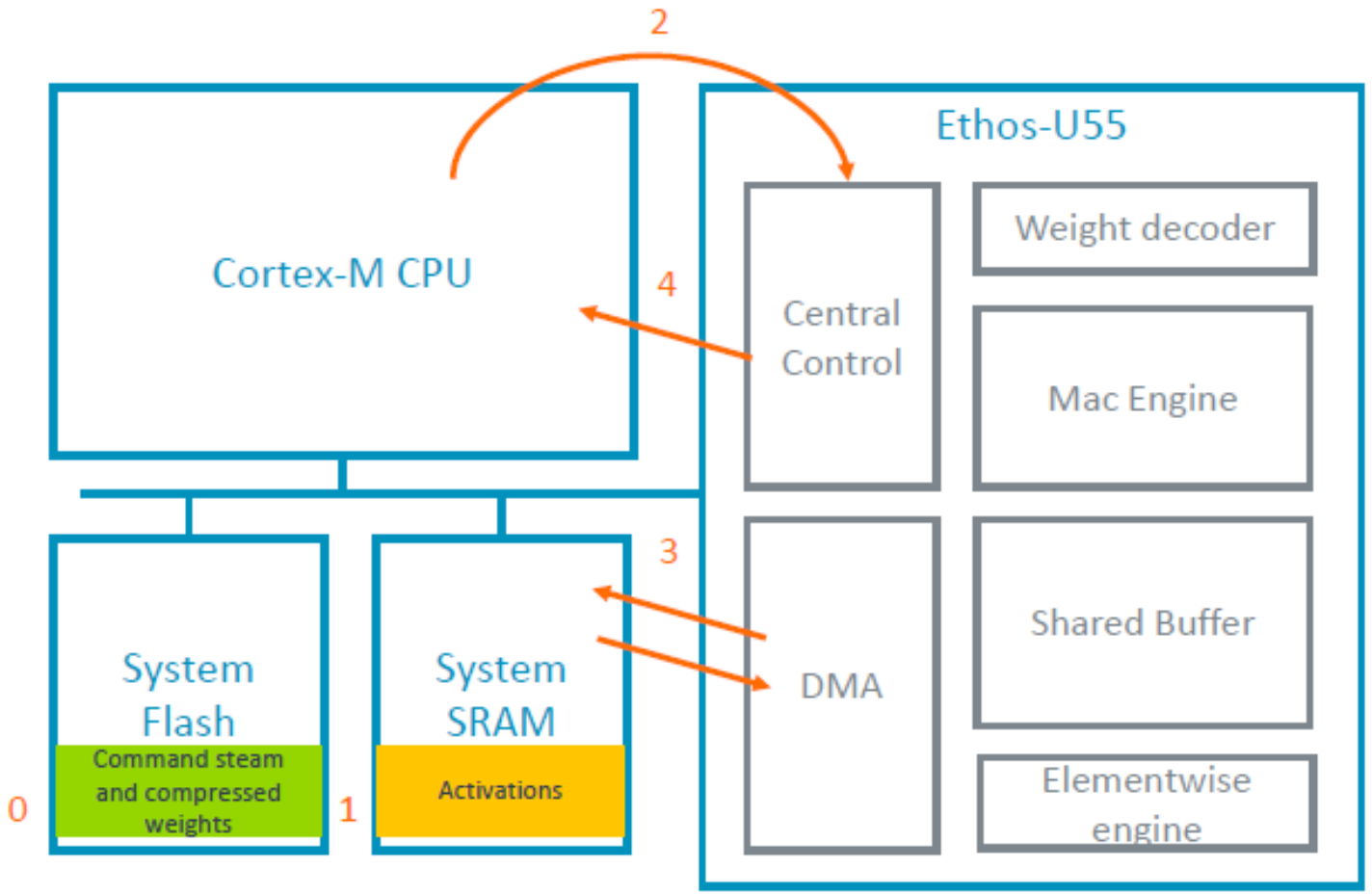


	Ethos-U55
Use Case	Inference only
MAC/Cycle	32/64/128/256 MACs/cycle 64GOPs-0.5TOPs@1GHz
Memory	SRAM + flash
CPU	Cortex-M
Bus	Two 64-bit AXI master infs M0: Full R+W AXI-M to SRAM M1: R only AXI-M to flash
Data type	8-bit input x 8-bit weights 16-bit input x 8-bit weights Weight is Compressed by Vela Compiler

arm

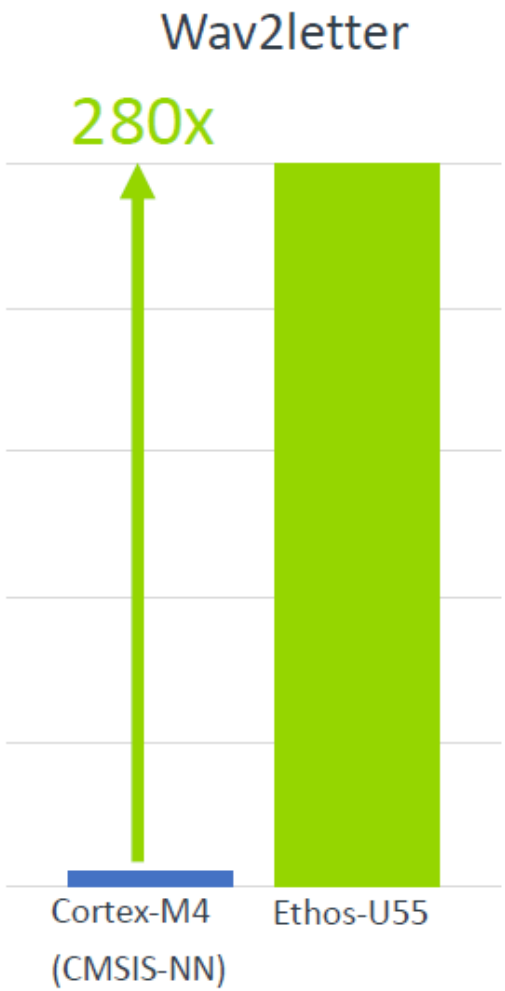
Typical Ethos-U55 Data Flow

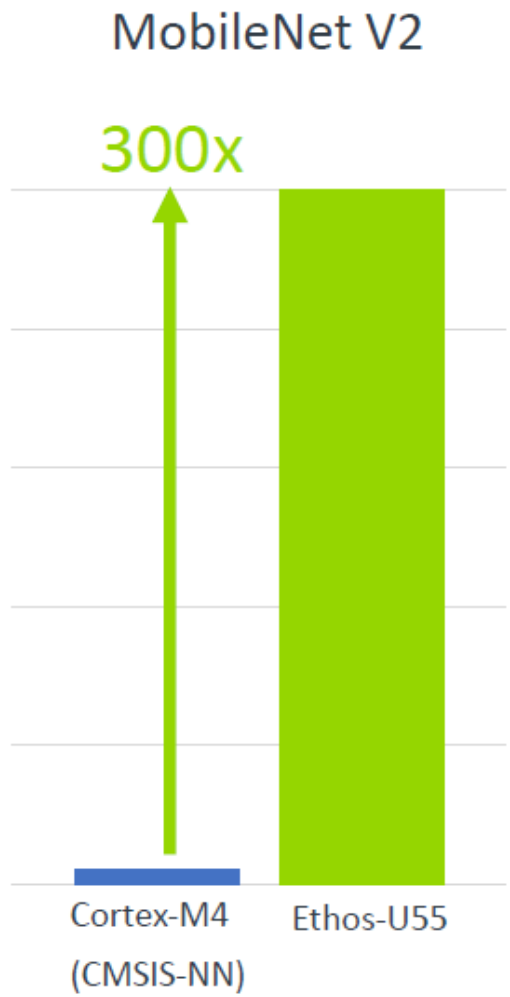
0. An offline compiled command stream with corresponding compressed weights are put into system Flash.
1. Input activations are put into system SRAM.
2. The host starts Ethos-U55 by defining all memory regions to be used, in particular the location of the command stream and input activations.
3. Ethos-U55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



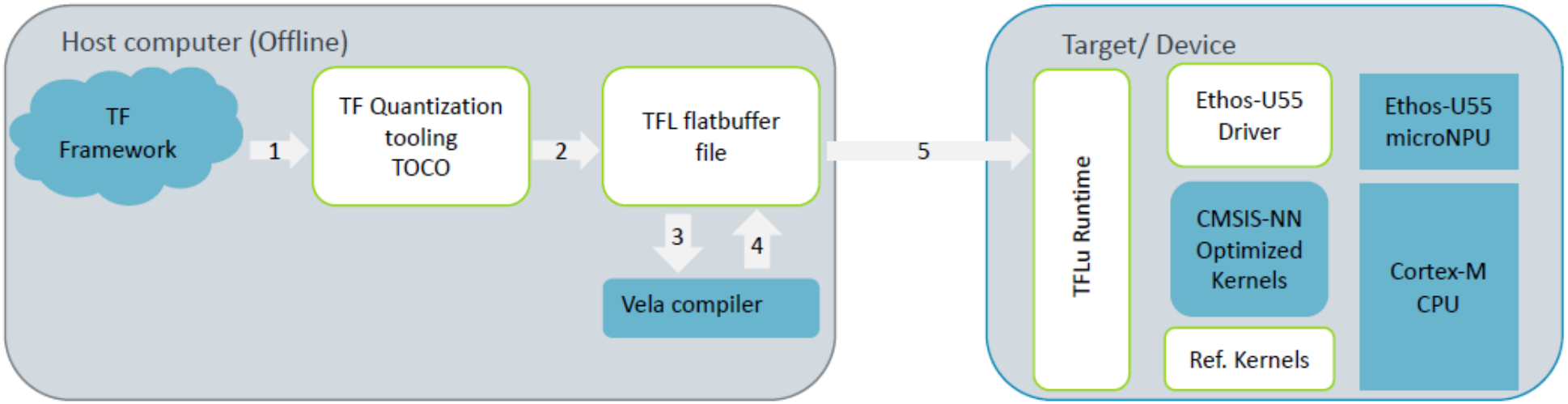
Ethos-U55 Performance Results

Using ***256 MACs/Cycle*** configuration vs. Cortex-M4 using CMSIS-NN optimizations





Ethos-U55 Optimized Software Flow



- Train network in TensorFlow
- Quantize it to Int8 TFL flatbuffer file (.tflite file)
- Vela compiler identifies graphs to run on Ethos-U55
  - Optimizes, schedules and allocates these graphs
  - Lossless compression, reducing size of tflite file

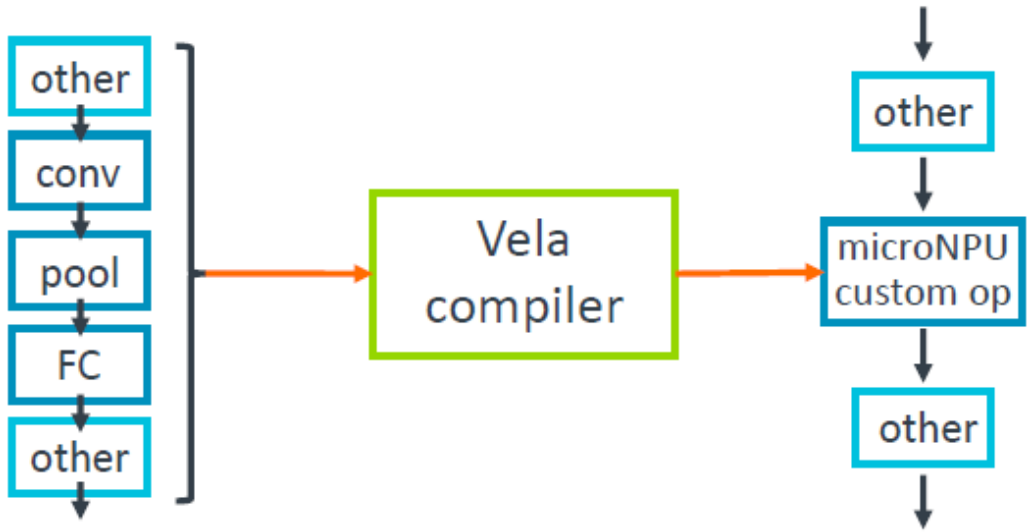
- Runtime executable file on device
- Accelerates kernels on Ethos-U55. Driver handles the communication
- The remaining layers are executed on Cortex-M
  - CMSIS-NN optimized kernels if available
  - Fallback on the TFLu reference kernels

arm

Vela Compiler

A Python based optimizer executed on your computer

- Reads a tflite file, writes a modified tflite file
- Generates commands for microNPU
- Optimizes scheduling of subgraphs
- Loss-less compression of weights
- Reduces SRAM and Flash footprint
- Enabling networks previously not feasible in embedded systems!
- Open source

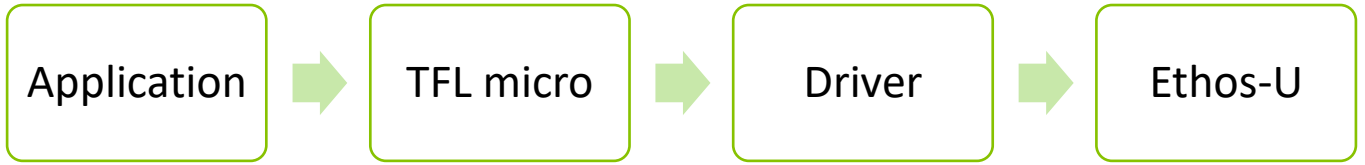


arm

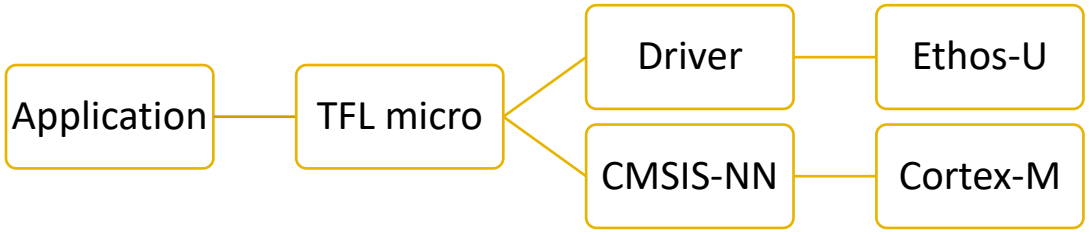
Network support in Ethos-U

- Ethos-U supports a fixed set of operators and can completely execute networks that map to that operator set. For ex:

  DS-CNN-L, ResNext 50, MobileNet v1, MobileNet v2, Inception v3, Inception v4, RNNoise, HAR_github, Deepspeech1, Wav2letter, … and many more



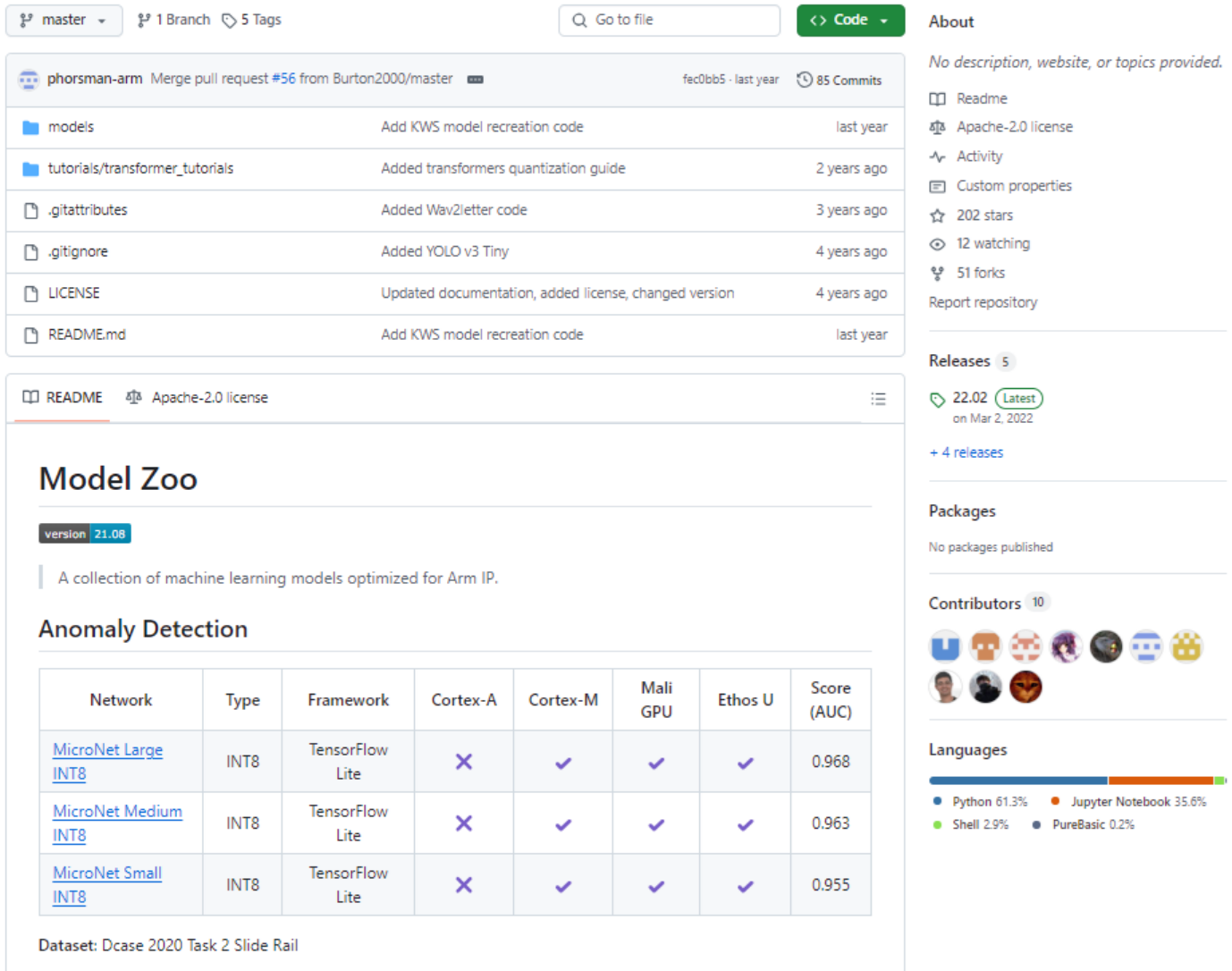
- For networks that cannot be executed on Ethos-U completely, the operators unsupported by Ethos-U fallback to the attached Cortex-M CPU
  - These are accelerated through CMSIS-NN library



https://review.mlplatform.org/plugins/gitiles/ml/ethos-u/ethos-u-vela/+/HEAD/SUPPORTED_OPS.md

arm

ML Model Zoo



master · 1 Branch · 5 Tags · Go to file · Code

phorsman-arm Merge pull request #56 from Burton2000/master — fec0bb5 · last year — 85 Commits

models	Add KWS model recreation code	last year
tutorials/transformer_tutorials	Added transformers quantization guide	2 years ago
.gitattributes	Added Wav2letter code	3 years ago
.gitignore	Added YOLO v3 Tiny	4 years ago
LICENSE	Updated documentation, added license, changed version	4 years ago
README.md	Add KWS model recreation code	last year

README · Apache-2.0 license

Model Zoo

version 21.08

> A collection of machine learning models optimized for Arm IP.

Anomaly Detection

Network	Type	Framework	Cortex-A	Cortex-M	Mali GPU	Ethos U	Score (AUC)
MicroNet Large INT8	INT8	TensorFlow Lite	✗	✓	✓	✓	0.968
MicroNet Medium INT8	INT8	TensorFlow Lite	✗	✓	✓	✓	0.963
MicroNet Small INT8	INT8	TensorFlow Lite	✗	✓	✓	✓	0.955

Dataset: Dcase 2020 Task 2 Slide Rail

About

No description, website, or topics provided.

Readme
Apache-2.0 license
Activity
Custom properties
202 stars
12 watching
51 forks
Report repository

Releases 5

22.02 Latest
on Mar 2, 2022

+ 4 releases

Packages

No packages published

Contributors 10

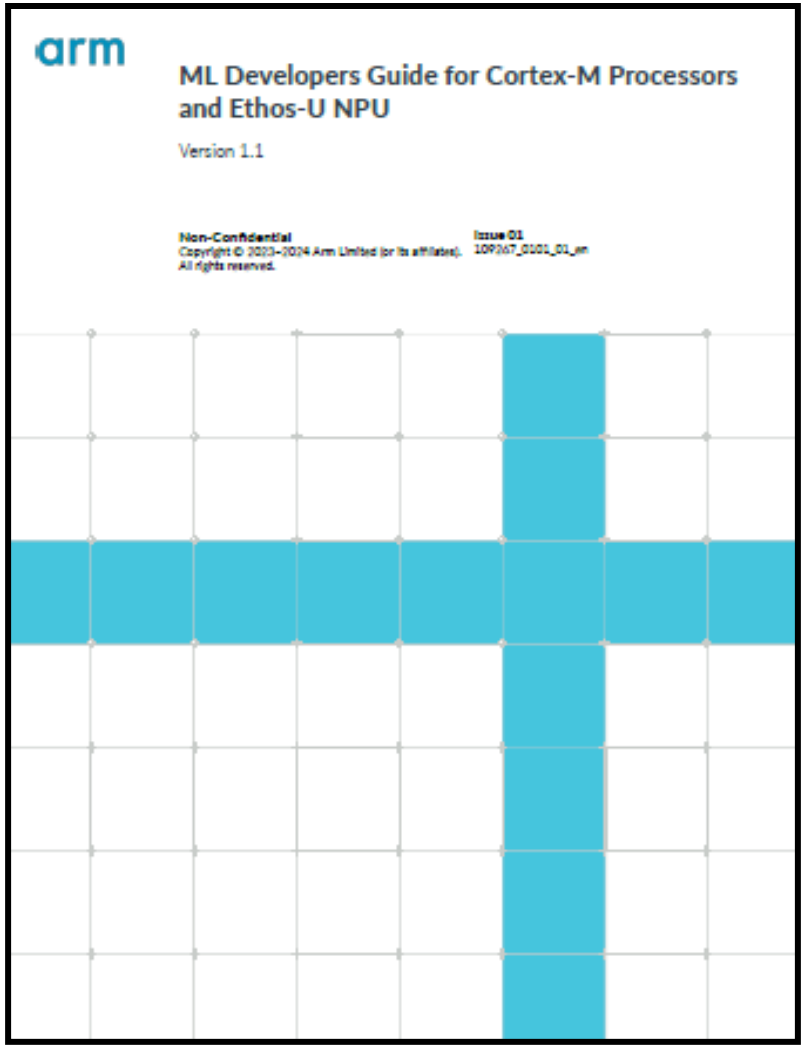
Languages

Python 61.3% · Jupyter Notebook 35.6% · Shell 2.9% · PureBasic 0.2%

https://github.com/ARM-software/ML-zoo

ML Developers Guide for Cortex-M Processors and Ethos-U NPU

1. Overview
2. ML software development for Arm Cortex-M processors
3. Arm Ethos-U NPU
4. Tool support for the Arm Ethos-U NPU
5. The Arm ML Zoo
6. ML Embedded Evaluation Kit
7. CMSIS-Pack based ML examples
8. Profiling and optimizing ML models
9. MLOps systems
10. Resources for Ethos-U



https://developer.arm.com/documentation/109267/latest/

Summary

- v8.1M architecture (Cortex-M55, M52 and M85) supports Helium (M profile Vector Extension).
- Helium is tuned to build DSP and ML application and CMSIS-DSP/NN can boost performance.
- Ethos-U accelerates Inference algorism
- Ethos-U55 target is Edge AI application
- Tensorflow is preferred as ML framework and Vela compiler optimize network for Ethos-U
- Some operators which is not supported by Ethos needs fall back to Cortex-M
- Please refer ML Developers Guide for Cortex-M Processors and Ethos-U NPU

arm

Thank You
Danke
Gracias
Grazie
谢谢
ありがとう
Asante
Merci
감사합니다
धन्यवाद
Kiitos
شكرًا
ধন্যবাদ
תודה
ధన్యవాదములు