

Introduction to the Arm Cortex-M55 Processor

By Joseph Yiu, Distinguished Engineer

February, 2020

arm

White Paper

Abstract

The Arm Cortex-M55 processor is Arm's most AI-capable Cortex-M processor and the first to feature Arm Helium vector processing technology, bringing enhanced, energy-efficient digital signal processing (DSP) and machine learning (ML) performance. This white paper provides an overview of the features of the Cortex-M55 processor, target applications, and how to get started with development.

Table of Contents

Topic

[2. Introduction](#)

[3. Overview](#)

[4. Technical Details](#)

[4.1 Processor](#)

[4.2 Floating-point Unit](#)

[4.3 Helium](#)

[4.3.1 Helium Support with the Cortex-M55 Processor](#)

[4.3.2 How Helium Helps Digital Signal Processing and Machine Learning](#)

[4.3.3 Performance of Helium](#)

[4.3.4 Additional Benefits of Helium](#)

[4.4 Memory System](#)

[4.5 Security](#)

[4.6 Debug](#)

[4.7 Innovation](#)

[5. Cortex-M55 Processor Applications](#)

[6. Software](#)

[7. Supporting IP](#)

[7.1 Corstone-300](#)

[7.2 The Ethos-U55 Processor](#)

[8. Conclusion](#)

Introduction

The Cortex-M55 processor is the first Arm Cortex-M processor supporting the [Armv8.1-M architecture](#). With [Helium technology](#) (also known as the M-Profile Vector Extension, MVE), Cortex-M55 based products can achieve a significant increase in performance and energy efficiency on signal processing and ML applications compared to previous Cortex-M based products. The Armv8.1-M architecture was announced during Embedded World 2019, and a white paper introducing Armv8.1-M can be found [here](#).

Apart from Helium technology, the Armv8.1-M architecture includes many other enhancements that bring additional benefits to the Cortex-M55 processor. There are a number of optional features at both the processor implementation and architectural levels to enable system-on-chip (SoC) designers to create designs that fit different requirements for their specific applications. This white paper explains these features in detail.

Overview

The Cortex-M55 processor is designed to deliver outstanding performance and energy efficiency for control, signal processing and ML with a small silicon footprint. Meanwhile, the design continues to align with the key requirements you will find in microcontrollers and embedded systems today, including:

- + Real-time capabilities
- + Security
- + Ease-of-use

Here is a quick summary of the key characteristics of the Cortex-M55 processor:

- + Armv8.1-M architecture with support for:
 - Optional Helium vector processing
 - Optional Floating-point Unit (FPU) with support for multiple floating-point formats
- + Optional TrustZone security extension
- + Advanced memory system features including caches and tightly coupled memory (TCM) support
- + Optional Secure and Non-secure Memory Protection Units (MPU) with up to 16 MPU regions for each of them
- + Up to 480 interrupts and non-maskable interrupt (NMI), with 8 to 256 levels of programmable priority levels
- + Optional coprocessor interface
- + Optional Arm Custom Instructions (available in 2021)
- + Various debug feature enhancements including new Performance Monitoring Unit (PMU)
- + 1.6 Dhrystone DMIPS/MHz and 4.2 CoreMark/MHz

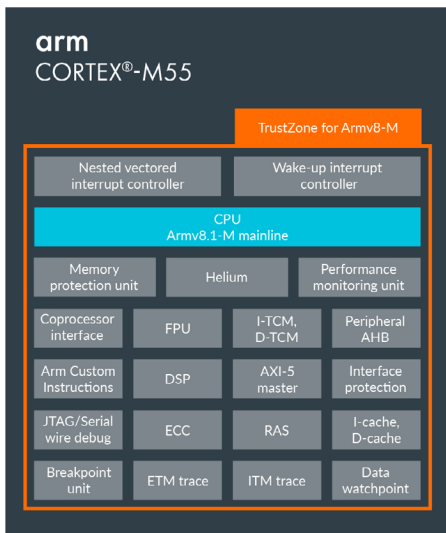


Fig. 1:
Cortex-M55
block diagram

Technical Details

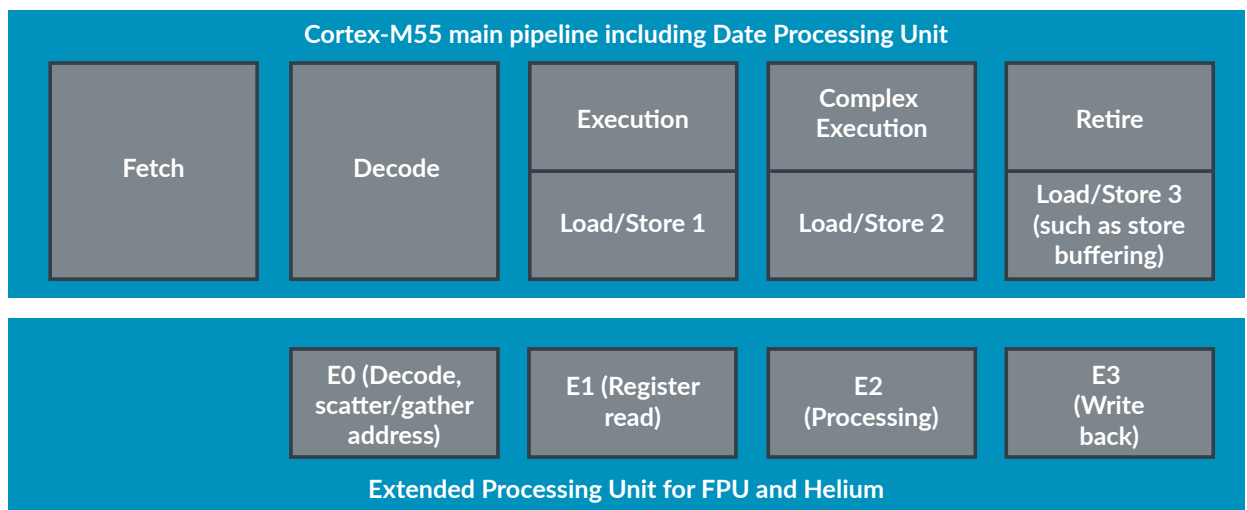
4.1 Processor

The processor in Cortex-M55 is based on a 4-stage integer pipeline design and when the Helium vector extension is included, the vector engine increases the total pipeline 5-stages. The pipeline is fully in-order (that is, no out-of-order execution) and a small amount of dual-issue capability is included. Two instructions can be issued at the same time when the instruction issuing stage detects that the next two instructions are both 16-bit, subject to the combination of instruction types. However, unlike the Cortex-M7 processor, the Cortex-M55 processor's dual issue capability is limited and is not classified as a superscalar processor. Still, this enables the Cortex-M55 to reach a performance of 1.6DMIPS/MHz, ~28% higher than the Cortex-M4 processor.

The pipeline is divided into two sides:

- ✦ The main pipeline, which is always present
- ✦ The extended processing unit, which is present only if FPU or Helium support is included

Fig 2:
Cortex-M55
pipeline



The separation of the pipeline allows the FPU or Helium unit to be powered down or placed into retention state if they are not being used.

The 4-stage pipeline enables the Cortex-M55 processor to have a modest increase in maximum clock frequency compared to the popular Cortex-M4 processor (typically over 10% depending on the configuration).

4.2 Floating-point Unit

The Cortex-M55 FPU support is based on Arm FPUv5 architecture which is fully IEEE-754 compliant. When the FPU is included, the Cortex-M55 processor supports scalar float-point instructions for data format of half-precision (16-bit, fp16), single-precision (32-bit, fp32), and double-precision (64-bit, fp64).

Support of half-precision floating-point arithmetic is new in Arm Cortex-M processors. In a range of sound processing and sensor data processing scenarios, a wide dynamic range is needed but the audio quality and signal resolution do not need to be high. In such applications, the half-precision floating-point format can be a good fit as we can process twice the amount of data per clock cycle when compared to using single-precision floats (32-bit), and at the same time reduce the memory footprint of data storage.

Single-precision float has been available in [Cortex-M processors](#) for quite a long time. When compared to the Cortex-M4 processor, single-precision floating-point support was enhanced (was FPUv4 in Cortex-M4, now FPUv5 in recent Cortex-M processors), and the performance of single-precision floating-point processing is significantly better. Because double-precision processing is relatively rare in microcontrollers and small IoT endpoints, the double-precision float support in the Cortex-M55 processor is focused on optimization for devices with a small area and low-power. But, by having native double-precision floating-point instruction support, the performance of such processing is still significantly higher than processors that do not support double-precision natively.

4.3 Helium

4.3.1 Helium Support with the Cortex-M55 Processor

Just like other Cortex-M processors, the Cortex-M55 processor is highly configurable and Helium support on the Cortex-M55 is also optional. From instruction-set support point of view, there are five combinations:

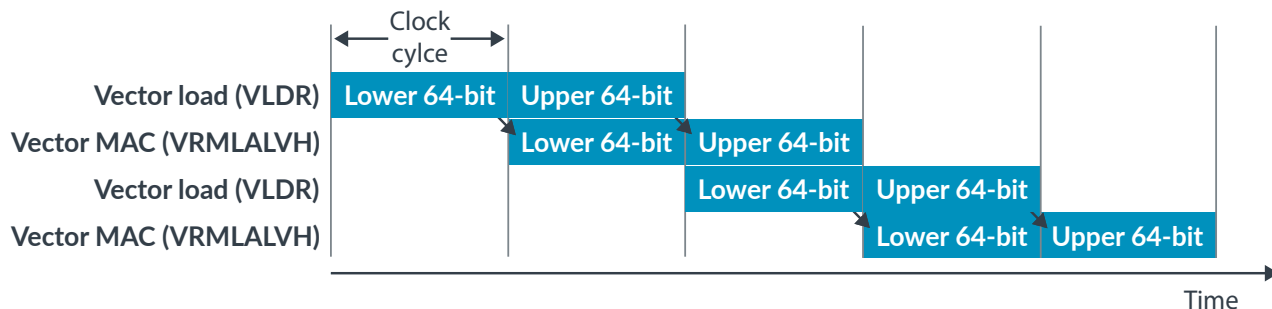
Config	FPU Data type: scalar float (fp16, fp32, fp64)	Helium Data type: vectored fixed-point (8-bit, 16-bit, 32-bit)	Helium Data type: vectored floating-point (fp16, fp32)
1	-	-	-
2	Included	-	-
3	-	Included	-
4	Included	Included	-
5	Included	Included	Included

These options allow SoC designers to customize the Cortex-M55 processor design to fit their specific application needs.

4.3.2 How Helium Helps Digital Signal Processing and Machine Learning

As explained in the [Armv8.1-M introductory white paper](#), Helium reuses the registers in the FPU as vector registers and each vector is 128-bit. The Cortex-M55 vector engine is implemented with a 64-bit internal data path, which is 2x the width of SIMD support in previous Cortex-M designs (32-bit). While each Helium operation takes two clock cycles, the architecture allows the Cortex-M55 to overlap execution cycles between instructions, doubling the performance for a range of code fragments where memory accesses and data processing can be carried out in parallel. This characteristic of the pipeline enables high energy efficiency by using multiple hardware resources simultaneously.

Fig. 3:
Vector data
processing

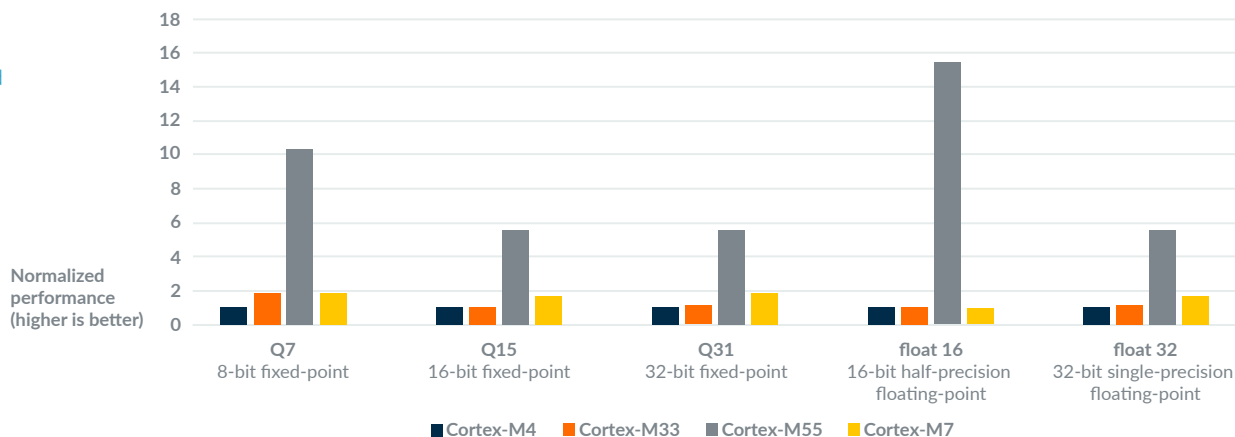


Meanwhile, new features like Low-overhead Branch Extensions and new vector memory access instructions allows further performance gains. As a result, the performance of the Cortex-M55 processor on vector data processing is over 4x when compared to previous Cortex-M4 processors. Such performance gains are well adapted to a range of signal processing algorithms like FIR filters, FFT, as well as ML processing tasks like inference using neural network (see next section 4.3.3).

4.3.3 Performance of Helium

A range of benchmark activities has been carried out in Arm during the development of Armv8.1-M and the Cortex-M55 processor. Starting from a high-level analysis based on application types, we see the Cortex-M55 delivers up to 5x improvement for DSP performance and up to 15x improvement for ML performance. For DSP workloads, we see various levels of performance uplift based on data types, as shown in figure 4.

Fig. 4:
Average performance
per datatype for selected
CMSIS-DSP Kernels Vs.
Cortex-M processors



4.3.4 Additional Benefits of Helium

Helium is not only useful for signal processing and ML applications. In many other applications that deal with data arrays, data processing could also be vectorized with auto-vectorization support in C compilers. As such, Helium can bring performance benefits to a wide range of general applications.

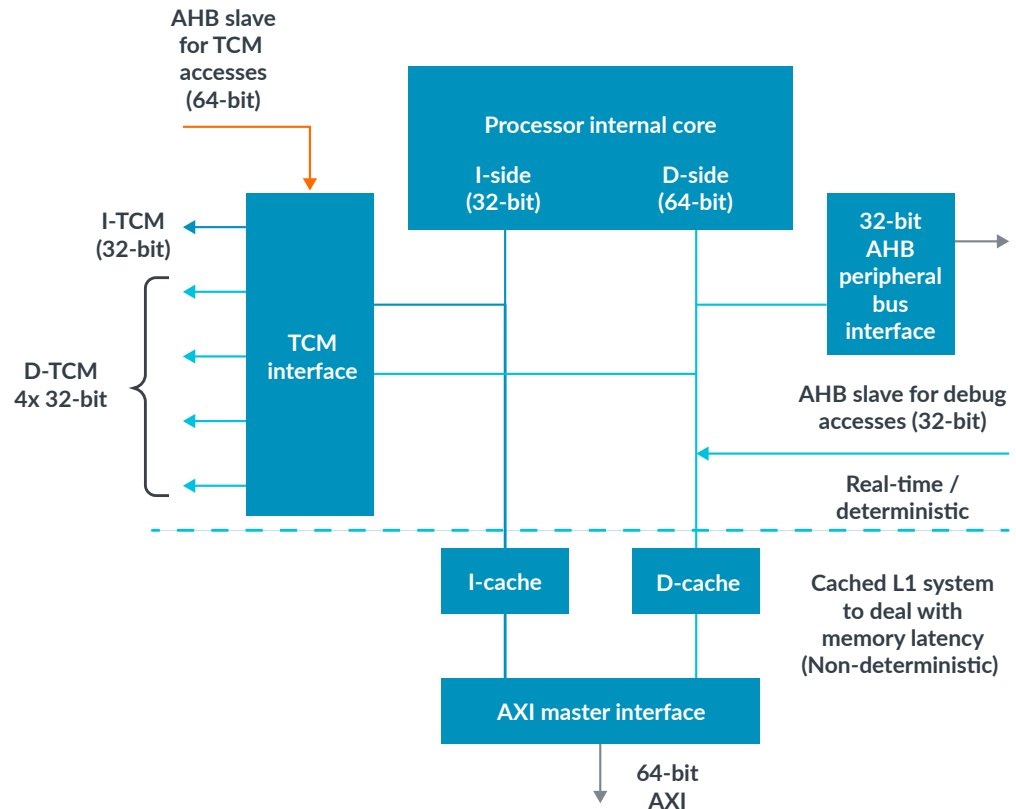
The Low-overhead Branch extension to the Armv8.1-M architecture avoids the need to do aggressive loop unrolling to get a high performance in certain situations. This enables applications to be compiled for high-level speed optimizations while keeping the code size small, enabling lower power and reducing costs. Some of the Low-overhead Branch instructions are available in Armv8.1-M even without Helium.

4.4 Memory System

The Cortex-M55 memory system is very similar to the one in the Cortex-M7 processor at a high level, however the details are different. The internal memory system is designed in two parts:

- ✦ A closely coupled part that is optimized for real-time, deterministic behaviors
- ✦ A cache-based bus system that enables the Cortex-M55 processor to be used with memory systems with higher latency

Fig. 5:
Cortex-M55
memory system



The key interfaces are described as follows:

- ✦ **AMBA 5 AXI main bus interface** – The 64-bit AXI5 interface allows for high bandwidth memory access and supports multiple outstanding transfers so that we can have a high-performance level even when the main memory (DDR) has a high latency. The AXI interface can also operate at a divided clock frequency speed from the processor clock. While the main bus interface has some new features from AXI5 (RAS extension, parity data checks, AWAKEUP for power control), only a subset of AXI5 features are implemented and therefore AXI4 bus interconnect IP can still be used with the Cortex-M55 processor.
- ✦ **Instruction cache** – The instruction cache is optional and can be configured from 0KB to 64KB when implemented. It is based on a 2-way set associative cache mechanism and supports optional error correction code (ECC).
- ✦ **Data cache** – The data cache is optional and can be configured from 0KB to 64KB when implemented. It is based on a 4-way set associative cache mechanism, supports Write-back (WB) and Write-through (WT) cache policies, and optional ECC.

- + **Instruction Tightly Coupled Memory (ITCM)** – The 32-bit instruction TCM is optional and can be configured from 0 to 16MB. It also supports wait-states and optional ECC.
- + **Data TCM** – The data TCM is optional and can be configured from 0 to 16MB. It also supports wait-states and optional ECC. Unlike the Cortex-M7 processor, the Cortex-M55 provides four 32-bit data TCM interfaces which split equally using bit[2] and bit[3] of the address value - so in total, the data -TCM interface supports up to 128-bit per cycle of data transfer bandwidth. Earlier we mentioned that the Helium data path inside the Cortex-M55 processor is 64-bit, so the processor software execution can only generate data traffic of 64-bit per cycle. However, in many signal processing or ML processing tasks, we also need to use direct memory access (DMA) operations to transfer new data into the data TCM and pull old results from the data TCM while the processor is running. Having the additional TCM interface and bandwidth allows the processor to handle those transfers simultaneously with software accesses. If both the software running on the processor and DMA controller tried to access the same data TCM memory bank, then software access is given higher priority, and it is likely that in the next clock cycle, software access will move on to another bank and then the DMA transfer can proceed.
- + **AHB slave interface for TCM** – This 64-bit AHB interface allows a DMA controller or other bus masters to access the instruction TCM and data TCM. Burst transfers are supported on this interface. The AMBA 5 AHB protocol is used as this arrangement has lower silicon area overhead compared to AXI, but a bus bridge component can be used to bridge AXI DMA controller to this AHB slave interface port easily.
- + **AHB peripheral bus interface** – The 32-bit AHB peripheral interface allows legacy AHB peripherals to be reused easily on the Cortex-M55 processor. In addition, it can reduce access latency by allowing peripheral register accesses to avoid the main AXI interconnect which might have some latency impact.
- + **Debug AHB** – The 32-bit debug AHB5 slave interface allows debug components like Debug Access Port (DAP) to access the memory system of the Cortex-M55 processor. Alternatively, a CoreSight debug subsystem can be used when a Cortex-M55 processor is used in a multi-core SoC design.

All the bus interfaces are based on bus protocols defined in the AMBA standard. These bus protocols are open, royalty-free and are proven in products in the market today. To help chip designers deal with system level integration, Arm also provides the Arm Corstone-300, a reference design that includes various system IP components for the Cortex-M55 processor. More details of these related system components are given later in this document.

4.5 Security

The [Arm TrustZone security](#) extension is supported in the Cortex-M55 processor and is a configurable option. This is because SoC designs might have other processors and the security-sensitive operations can be handled somewhere else.

Armv8.1-M introduced several security enhancements including a new MPU region attribute called Privileged eXecute Never (PXN), Unprivileged Debug Extension (UDE) and some enhancements in relation to TrustZone (instructions such as CLRM and VSSCLRM which can clear Secure data from multiple registers quickly).

With the new features available in Armv8.1-M, it is possible to have isolated debug permission of different software components in each security domain in the Cortex-M55 processor. For example, a silicon vendor might need to include third-party libraries in their Secure firmware. This new capability allows the silicon vendor to restrict the debug visibility into the unprivileged library under development. This allows third-party developers to develop the software, but are not able to reverse engineer the privileged Secure firmware from the silicon vendor, or other unprivileged Secure software components already preloaded on the devices.

4.6 Debug

The Cortex-M55 processor supports a range of debug features that are already available in most other Cortex-M processors, including:

- + Halt mode and monitor mode debug with on-the-fly debug access to memory space
- + Up to 8 hardware breakpoints, and unlimited software breakpoints
- + Up to 4 data watchpoints
- + Instruction trace with Embedded Trace Macrocell (ETM)
- + Selective data trace, event trace and profiling trace using Data Watchpoint and Trace Unit (DWT)
- + Software generated trace using an Instrumentation Trace Macrocell (ITM)
- + Debug authentication interface supporting TrustZone

There are also other new debug enhancements:

- + **Performance Monitoring Unit** – the DWT has been extended to include PMU functionality. This includes eight 16-bit event counters for counting architectural and implementation events and these counters can be cascaded in pairs if needed. Development tools, such as [Arm Streamline Performance Analyzer](#), will be able to use this feature to provide a detailed performance analysis of software.
- + **Direct cache access registers** – These registers allow the cache states (tag) to be accessed and development tools like [Arm Development Studio](#) will be able to use this feature to provide [Cache Data View](#).
- + **Unprivileged Debug Extension (UDE)**

The Cortex-M55 design bundle includes a debug access port module (for JTAG and Serial Wire Debug interface) and Trace Port Interface Unit (TPIU). The processor design support is also fully CoreSight compatible. To use the Cortex-M55 processor in a multi-core system design, chip designers can link up the debug system of the Cortex-M55 processor with other debug systems in the chip using solutions like [CoreSight SoC-600](#) and [Coresight SoC-600M](#). That allows the debugger to access the debug and trace features of multiple processors and other IP using a single debug and trace connection.

4.7 Innovation

The Cortex-M55 processor supports the same coprocessor interface that was introduced in the Cortex-M33 and Cortex-M35P processors. Existing hardware accelerators designed for these processors can be reused on the Cortex-M55 processor straight away. Using the coprocessor interface, SoC designers can create closely coupled hardware accelerators to speed up a certain range of processing functions.

A future release of the Cortex-M55 processor will support Arm Custom Instructions, providing another way to speed-up specialized data processing functions (available in 2021). [Click here for more details about Arm Custom Instructions.](#)

Cortex-M55 Processor Applications

The Cortex-M55 processor is intended for use in a wide range of applications. In addition to traditional microcontroller markets (such as consumer IoT, industrial and motor control, robotics, medical and fitness devices), where Cortex-M processors are very popular, there are many new application segments where the Cortex-M55 processor with Helium can bring significant benefits:



Earbuds



Fingerprint
unlock



Health
trackers



Predictive
maintenance



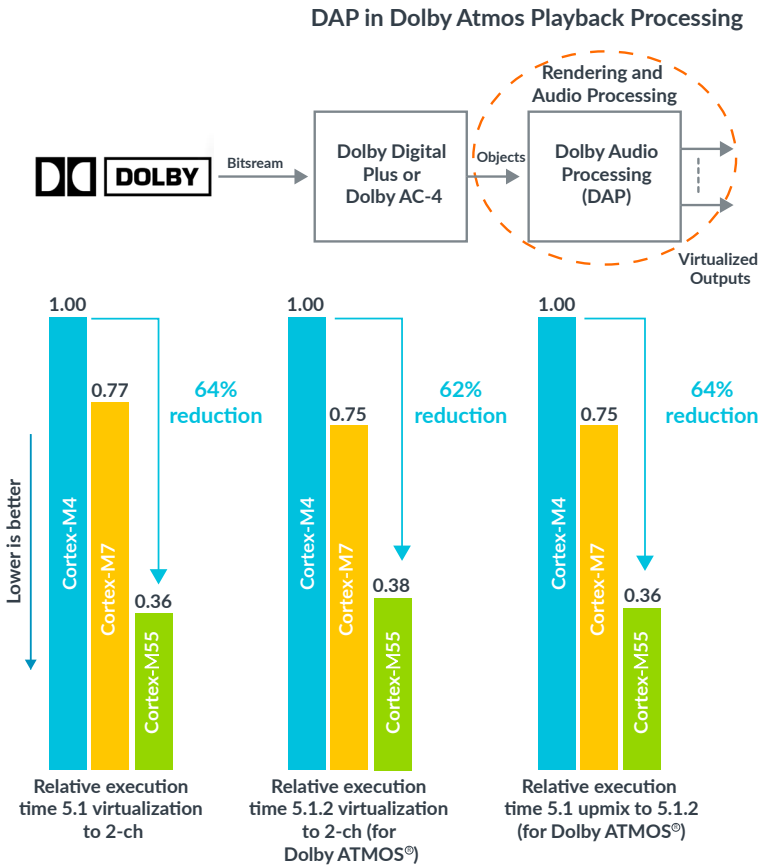
Smart
speakers



Video
doorbell

For example, Arm has worked closely with Dolby into their investigation of using the Cortex-M55 for Dolby audio processing. From the analysis results, we see that the Cortex-M55 processor can provide over a 60% reduction in execution time when compared to the Cortex-M4 processor.

Fig. 6:
Performance results
for the Dolby Atmos,
comparing the
Cortex-M55 processor
with previous generation
Cortex-M and Cortex-A
processors



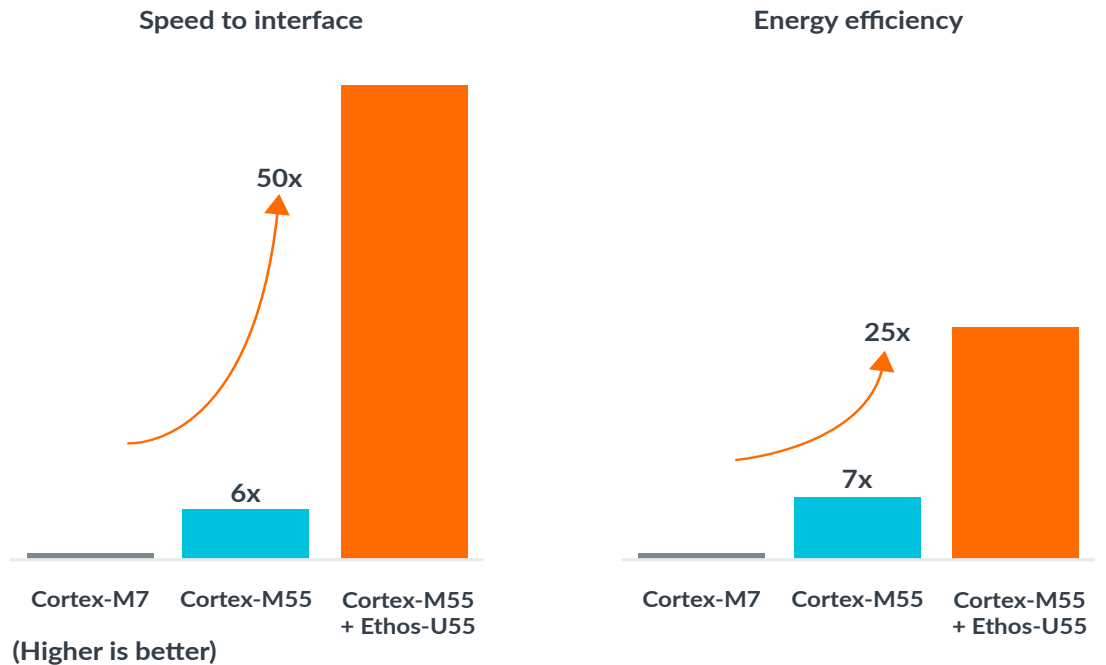
By using the Cortex-M55 processor, product designers can create audio products that are Dolby ATMOS capable at much greater efficiency and with a lower cost.

For sensor fusion processing workloads, we see a great improvement in processing time compared to previous Cortex-M processors. For example, an analysis carried out by Hillcrest Labs on quaternion multiplication kernel (for motion detection processing) running on the Cortex-M55 model shows a significant speed-up gained when using Helium technology. Instructions with only a single number of quaternions are increased by up to 2.5 times, and instructions with eight or more are increased by at least 4 times. For more information [download this white paper](#) by Hillcrest Labs and Arm.

NoQ	Cortex-M4 Ratio	Cortex-M7 Ratio	Cortex-M4 Speedup	Cortex-M7 Speedup
1	30.39%	44.93%	3.3x	2.3x
8	17.86%	27.68%	5.6x	3.7x
16	17.09%	26.87%	5.9x	3.8x
32	16.69%	26.45%	6x	3.8x

For ML applications, the performance gain is even more significant. Based on analysis carried out by Arm research, the Cortex-M55 processor can deliver a 6x performance boost when compared to the Cortex-M7 processor in voice assistant applications. If even higher ML performance is required, the Ethos-U55 processor, which is detailed in section 7.2, could be ideal as a companion accelerator for the Cortex-M55 processor.

Fig. 7:
Typical workloads for a voice assistant comparing the Cortex-M55 and Ethos-U55 processors to the Cortex-M7 processor (for more information about the Ethos-U55, see section 7.2).



Software

While the Cortex-M55 processor can deliver outstanding signal processing and neural network inference capability, we need software developers to deliver the software. Fortunately, as the Cortex-M55 processor is based on the same architecture series used by millions of embedded software developers today, it is very easy to use and many existing applications can be ported to the Cortex-M55 processor easily.

- + With the advancement in compiler technologies, many applications can take advantage of Helium technology by just upgrading the C compilers and enable Helium in the project options.
- + [CMSIS-DSP](#) with Helium support is available now and software developers can gain the performance benefit by swapping the CMSIS-DSP with the Helium-enabled version. Meanwhile, new functions are being added to the CMSIS-DSP library to allow Arm Cortex processors to be used in even more compute-intensive applications.
- + [CMSIS-NN](#) (Neural network) library will also be updated to support Helium technology. The CMSIS-NN libraries are tightly integrated into ML software frameworks like TensorFlow Lite micro.
- + [Trusted Firmware-M](#) is being updated to support the Cortex-M55 processor and Corstone-300, a system IP package (see section 7.1).

- ✦ In regards to ML frameworks, [TensorFlow Lite Micro](#) is fully supported by the Cortex-M55 and Ethos-U55 toolchain. The drivers for these processors will automatically optimize developers' TensorFlow models for any hardware configuration they wish to deploy. Learn more in section 7.2 below.
- ✦ Additional software enablement activities are ongoing with a variety of algorithm, software, tools and RTOS partners to deliver optimized software libraries that speed up time to development.

To get started today, the Cortex-M55 processor is supported by Arm Compiler 6.14, available in MDK v5.30 and Arm Development Studio. A Cortex-M55 fixed virtual platform (FVP) is available free-of-charge for software developers. A configurable Cortex-M55 Fast Model with SystemC interfaces supports custom virtual prototype designs. Learn more about Arm tools for Cortex-M55 [here](#).

Fig. 8:
Arm's extensive AI partner ecosystem of silicon, algorithm, software, tools and RTOS partners



Supporting IP

7.1 Corstone-300 Reference Design

To enable SoC designers to create Cortex-M55 based designs quickly, Arm has developed Corstone-300, one of the Corstone packages that provide a range of system IP components as well as a reference system design. Together with associated software and tools support, Corstone-300 is a solution to reduce cost and risks for creating secure systems.

Additionally, the Arm Artisan Physical IP libraries provide a low-power, integrated end-to-end IoT solution for Corstone-300 based SoC implementations.

The Corstone-300 reference design integrates the Cortex-M55 processor with an optimized AMBA AXI-based system bus. It demonstrates implementation of TrustZone for Armv8-M over AMBA AXI and shows integrated power control throughout the system.

The IP includes several useful components such as:

- ✦ A range of TrustZone security management IP such as CoreLink SIE-200 and CoreLink SIE-300

- ✦ CoreLink NIC-400-Lite configurable AXI interconnect
- ✦ A range of AMBA AXI and AHB5 bridges, including components for bridging between AXI and AHB5
- ✦ CoreLink Power Control Kit (PCK-600)
- ✦ Generic Flash Controllers (CoreLink GFC-100 and GFC-200)
- ✦ True Random Number Generator (TRNG) and Real-Time Clock (RTC)

The Corstone-300 reference design gives silicon vendors a jumpstart and it is easily customizable for a broad range of use cases. Corstone-300 platforms will be supported in open-source software, such as Trusted Firmware-M and Amazon FreeRTOS, enabling Arm partners to easily port their software. Corstone-300 is driven by a system architecture designed with TrustZone security. Together with associated software, Corstone-300 accelerates the route to PSA Certified silicon and devices.

7.2 The Ethos-U55 Processor

The Ethos-U55 is Arm's first microNPU, a new ML processor specifically designed to accelerate ML inference in cost-sensitive, area-constrained IoT and embedded devices. It is designed to fit into Cortex-M based systems to offload neural network tasks. It is configurable from 32 to 256 MAC per cycles. With software support for the TensorFlow ML framework, the Ethos-U55 is ideal for ML acceleration in applications that require speech recognition, keyword spotting and image classification.

The Ethos-U55 processor is designed with two AMBA AXI master interfaces, an APB interface for programming of configuration and control registers, interrupt signal for signaling to the host processor and power management control signals. The two AMBA AXI master interfaces are 64-bit – one is for read/write and the other is read-only for access data in flash. If all data is in SRAM, the read-only interface can be tied off and not used.

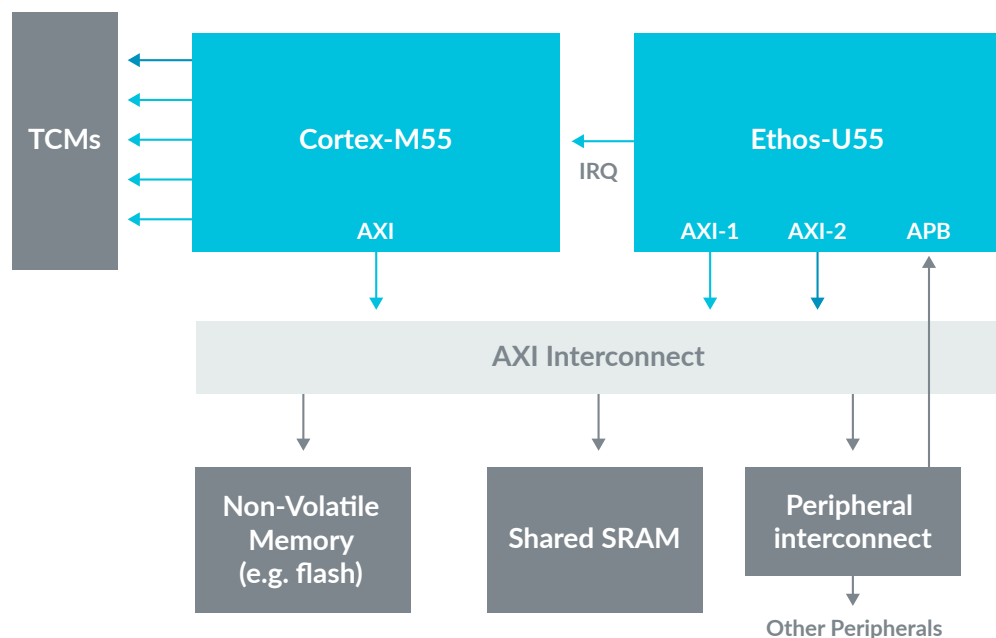
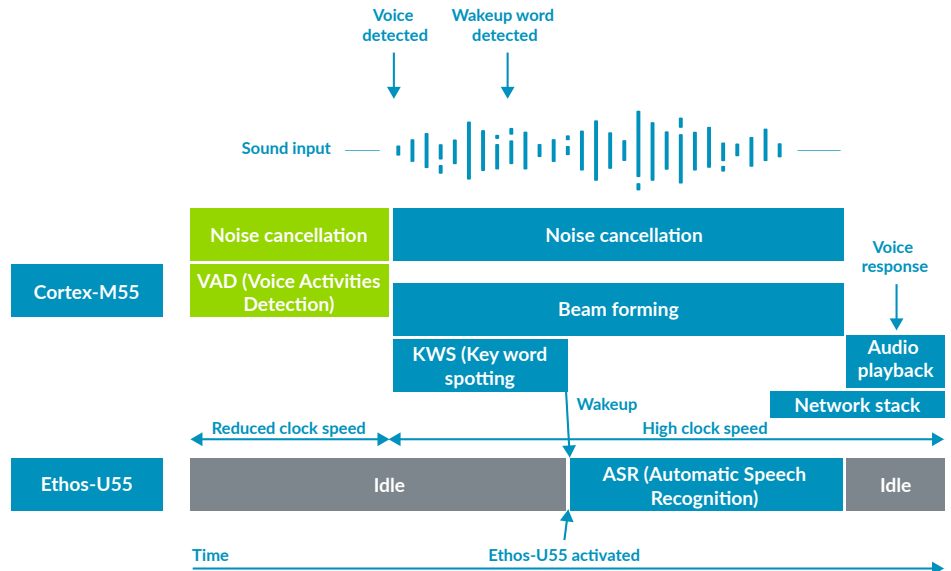


Fig. 9:
System of the Cortex-M55
and Ethos-U55 processors

The AMBA AXI interfaces on Ethos-U55 are 64-bits wide and the second AMBA AXI provides a dedicated AMBA AXI channel for accessing data in the non-volatile memory in typical microcontroller systems. For the majority of MCU applications, the command lists for Ethos-U55 are precompiled and placed in flash memories, and the Cortex-M processor can kickstart the neural networks processing by issuing start command and command list pointer via the APB control interface. When the processing is completed, the Ethos-U55 processor issues an interrupt event back to the Cortex-M processor.

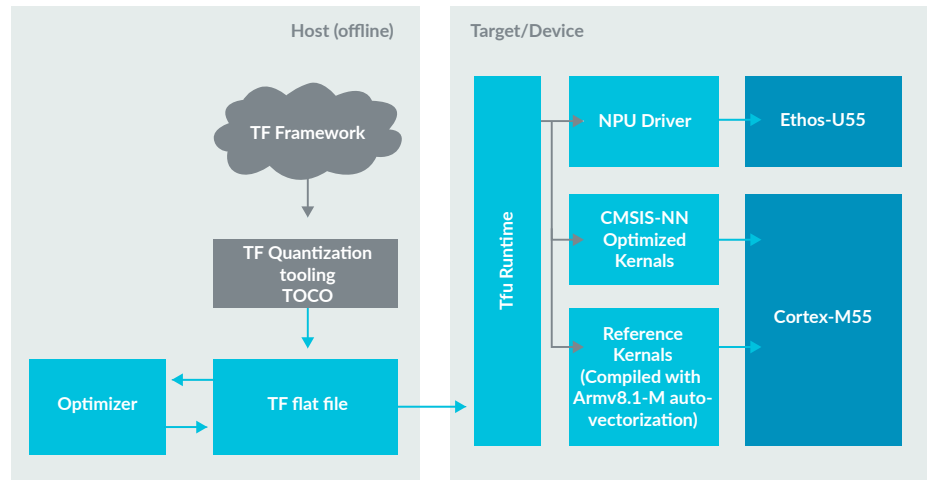
In a smart speaker application, the Cortex-M55 and Ethos-U55 processors work nicely together. For example, by default the Ethos-U55 can stay in a low-power mode while the Cortex-M55 is used to detect voice and wake-up word. Once the wake-up word is detected, the Ethos-U55 can perform the neural network processing in the ASR (Automatic Speech Recognition).

Fig. 10:
The Cortex-M55 and
Ethos-U55 processors
working together in a
smart speaker application



Software developers can benefit from the processing capability of Ethos-U55 by using the TensorFlow ML framework. After the TensorFlow model has been quantized into a TensorFlow Lite (TFL) model, the TFL FlatBuffer file is then inspected using an optimizer tool from Arm. The tool identifies which ML operators can be processed by the Ethos-U55 microNPU and substitutes these with a sequence of special operations; other ML operators may be processed on the Cortex-M processor by optimized kernels from the CMSIS-NN library. In the unlikely event that an ML operator is unavailable in both the Ethos-U55 microNPU and the CMSIS-NN library, then processing of that operator will fall back to use the reference implementation. The reference implementation and CMSIS-NN library are both able to take advantages of Helium technology by using advanced optimizations in C/C++ compilers to enable auto-vectorization, or using other instructions introduced in the Armv8.1-M architecture.

Fig. 11:
The Cortex-M55 and
Ethos-U55 processors
using the TensorFlow
ML framework



Conclusion

The Cortex-M55 processor is Arm's most AI-capable Cortex-M processor and the first to feature Arm Helium vector processing technology. Based on the same design principles of the Cortex-M family, the processor:

- ✦ Enhances endpoint AI performance bringing the highest, most efficient, real-time ML and DSP performance for Cortex-M
- ✦ Differentiates your design by using the coprocessor interface or by integrating Arm Custom Instructions to extend processor capabilities for specific workload optimization (available in 2021)
- ✦ Accelerates time to market with the Corstone-300 reference design with TrustZone, simplifying security and accelerating the route to PSA Certified silicon and devices
- ✦ Simplifies software development with a single developer toolchain supported by a broad ecosystem of software, tools, libraries and resources

With the addition of Helium technology, the Cortex-M55 processor achieves a significant performance uplift in signal processing and ML applications in the small footprint of a Cortex-M processor. In addition, the Armv8.1-M architecture can also help boost performance for standard applications where some of the data processing operations can be vectorized, and where some of the new branches, loops and conditional execution instructions can be utilized to enable better performance and smaller code size.

In addition to performance enhancements, there is also a range of new features including enhancements in security and new features in debug. With the coprocessor interface and support for Arm Custom Instructions, the Cortex-M55 processor is ideal for many low-power embedded and IoT applications where performance, energy efficiency and security are all needed.

For even more demanding ML systems, the Cortex-M55 can be easily paired with the Ethos-U55, as it is fully integrated into a single Cortex-M toolchain, delivering a 480x performance uplift in ML performance over existing Cortex-M processors.

With the backing of a strong ecosystem and delivering a 480x performance uplift and various supporting projects like CMSIS-DSP, CMSIS-NN and Trusted Firmware-M, getting started on application development with the Cortex-M55 processor is as easy as using previous Cortex-M processors.

For more information about the Cortex-M55 processor, supporting IP and related tools and software, visit the links below.

Reference

[Cortex-M55 web page](#)

[Corstone-300 web page](#)

[Ethos-U55 web page](#)

[Arm Helium technology web page](#)

[Arm TrustZone technology web page](#)

[Arm Custom Instructions web page](#)

[Introduction to the Armv8.1-M architecture white paper](#)

[Keil MDK web page](#)

[Fast Models and Fixed Virtual Platforms](#)

[TensorFlow Lite Micro](#)

[CMSIS](#)

[Arm Development Studio web page](#)

[Trusted Firmware website](#)

[Platform Security Architecture \(PSA\) website](#)

[Artisan Physical IP Libraries web page](#)



All brand names or product names are the property of their respective holders. Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given in good faith. All warranties implied or expressed, including but not limited to implied warranties of satisfactory quality or fitness for purpose are excluded. This document is intended only to provide information to the reader about the product. To the extent permitted by local laws Arm shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information.

© Arm Ltd. 2020