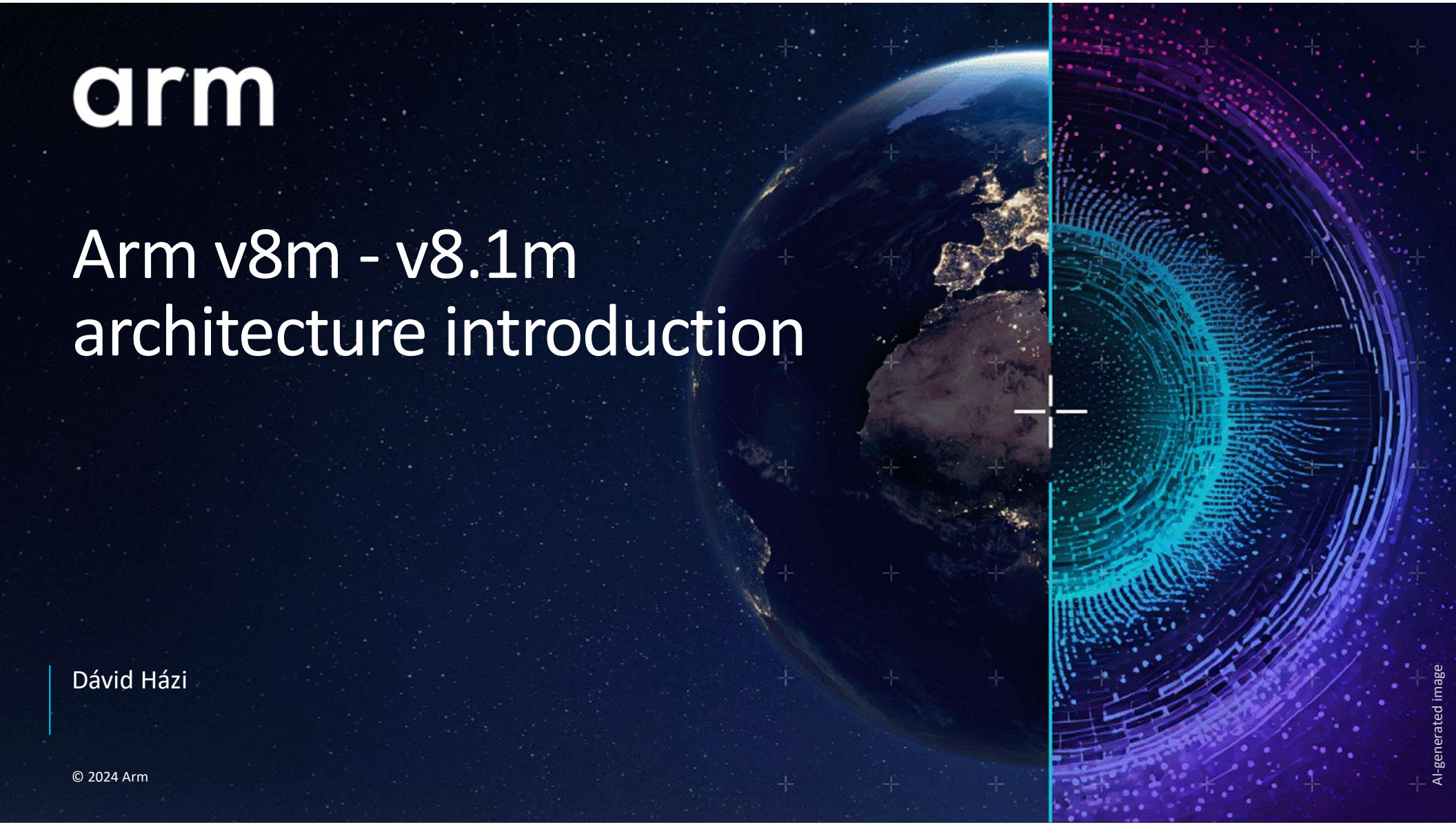


Arm v8m - v8.1m architecture introduction

Dávid Házi

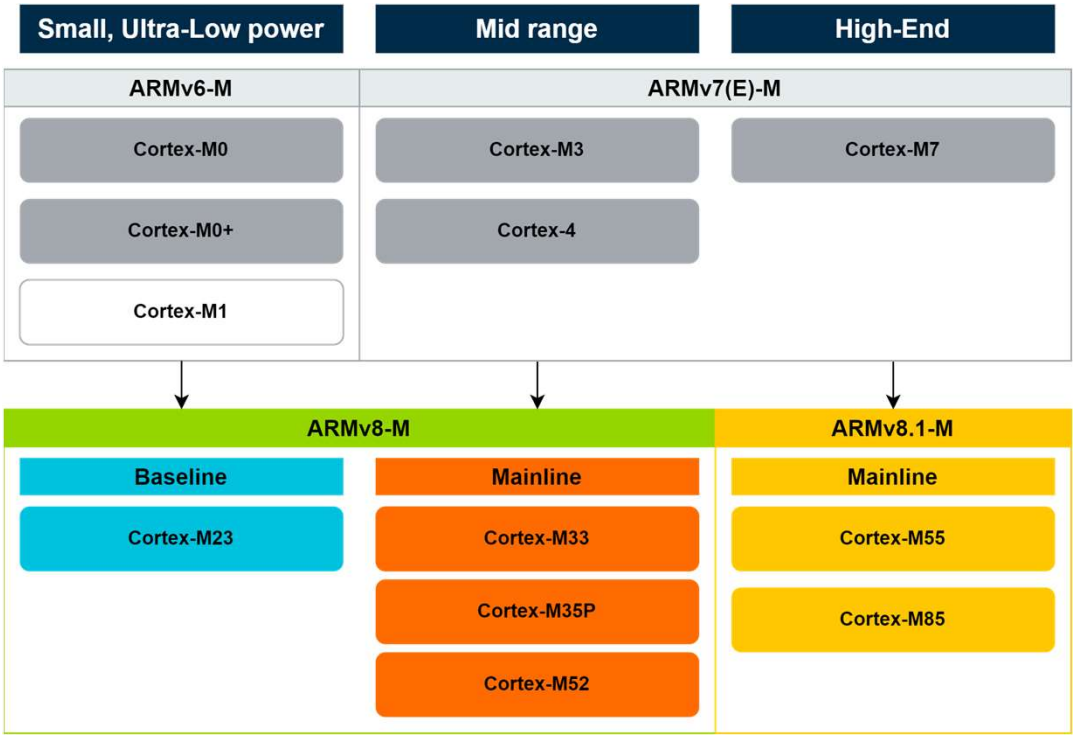
AI-generated image

Agenda

- Arm introduction
- v8m/v8.1m architecture overview
- MPU
- Subsystems
- Trustzone
- Trusted Firmware M
- “Real” life example, demo

Armv8m/Armv8.1m – What has changed? What's new?

- The next generation of ARM Cortex-M processors will be powered by **ARMv8-M** architecture



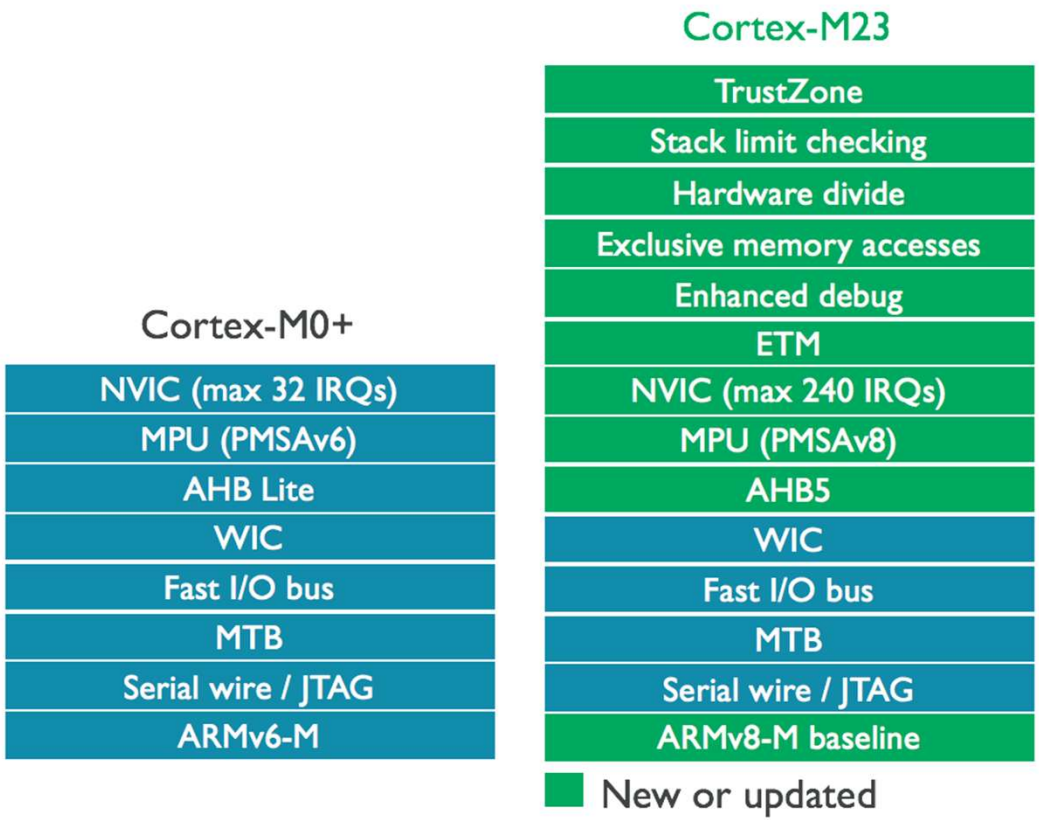
- Arm Cortex-M Processor Comparison Table

Armv8m/Armv8.1m – What has changed? What's new?

Feature	Cortex- M0	Cortex-M0+	Cortex- M1	Cortex- M23	Cortex- M3	Cortex- M4	Cortex- M33	Cortex- M35P	Cortex-M52	Cortex-M55	Cortex-M7	Cortex-M85
Instruction Set Architecture	Armv6-M	Armv6-M	Armv6-M	Armv8-M Baseline	Armv7-M	Armv7-M	Armv8-M Mainline	Armv8-M Mainline	Armv8.1-M Mainline	Armv8.1-M Mainline	Armv7-M	Armv8.1-M Mainline
TrustZone for Armv8-M	No	No	No	Yes (option)	No	No	Yes (option)	Yes (option)	Yes (option)	Yes (option)	No	Yes
Helium (M-Profile Vector Extension)	No	No	No	No	No	No	No	No	Single-beat (option)	Dual-beat (option)	No	Dual-beat (option)
PACBTI Extension	No	No	No	No	No	No	No	No	Yes (option)	No	No	Yes (option)
Floating-Point Unit (FPU)	No	No	No	No	No	SP (option)	SP (option)	SP (option)	HP, SP, DP (option)	HP, SP, DP (option)	SP, DP (option)	HP, SP, DP (option)
Digital Signal Processing (DSP) Extension	No	No	No	No	No	Yes	Yes (option)	Yes (option)	Yes	Yes	Yes	Yes
Hardware Divide	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Arm Custom Instructions	No	No	No	No	No	No	Yes (option)	No	Yes (option)	Yes (option)	No	Yes (option)
Coprocessor Interface	No	No	No	No	No	No	Yes (option)	Yes (option)	Yes (option)	Yes (option)	No	Yes (option)
DMIPS/MHz*	0.96	0.99	0.88	1.03	1.24	1.26	1.54	1.50	1.60	1.69	2.31	3.13
CoreMark®/MHz*	2.33	2.46	1.83	2.64	3.45	3.54	4.10	4.10	4.30	4.40	5.29	6.28
Maximum # External Interrupts	32	32	32	240	240	240	480	480	480	480	240	480
Maximum MPU Regions	0	8	0	16	8	8	16	16	16	16	16	16
Main Bus	AHB Lite (32-bit)	AHB Lite (32-bit)	AHB Lite (32-bit)	AHB (32-bit)	AHB Lite (32-bit)	AHB Lite (32-bit)	AHB (32-bit)	AHB (32-bit)	AXI (32-bit) or AHB (32-bit)	AXI (64-bit)	AXI (64-bit)	AXI (64-bit)
Instruction Cache	No	No	No	No	No	No	No	2-16kB	0-64kB	0-64kB	0-64kB	0-64kB
Data Cache	No	No	No	No	No	No	No	No	0-64kB	0-64kB	0-64kB	0-64kB
Instruction TCM	No	No	0-1MB	No	No	No	No	No	0-16MB	0-16MB	0-16MB	0-16MB
Data TCM	No	No	0-1MB	No	No	No	No	No	0-16MB	0-16MB	0-16MB	0-16MB
Dual Core Lock-Step (DCLS) Configuration	No	No	No	No	No	No	No	Yes	Yes (option)	Yes (option)	Yes (option)	Yes (option)

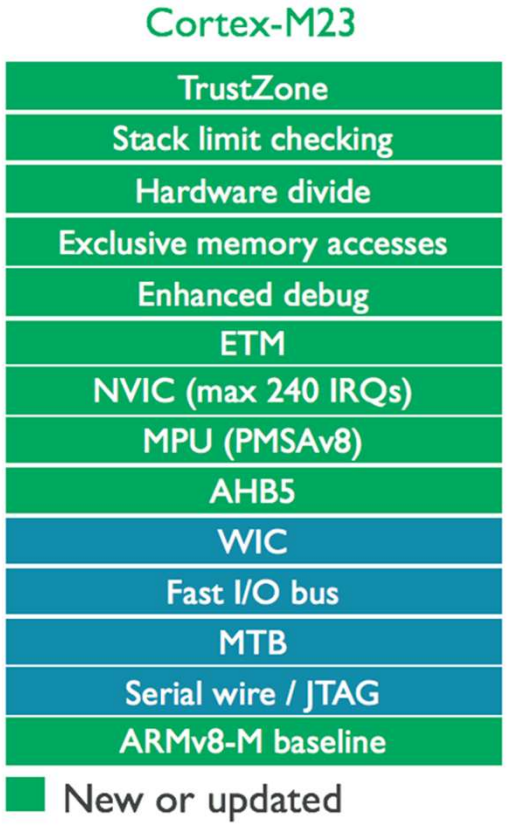
arm

Armv8m baseline



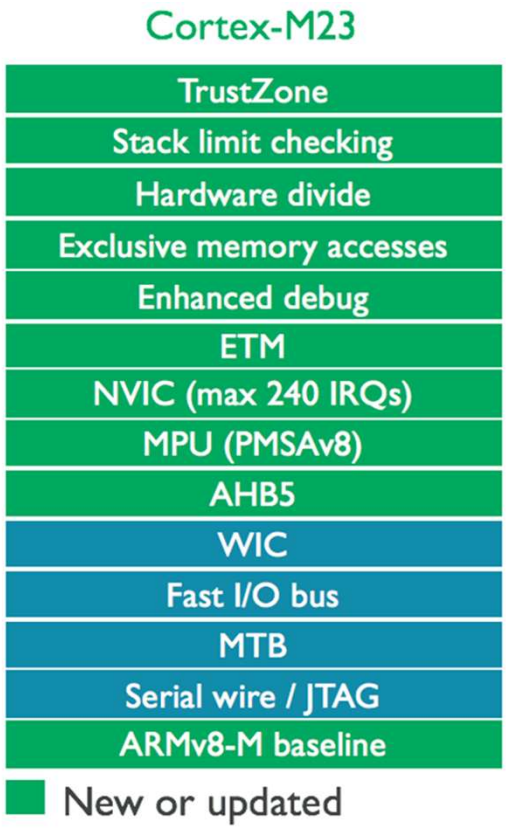
Armv8m baseline Cortex-M23

- Cortex-M23 implements the ARMv8-M architecture. Full details here.
- Uses
  - Same debug interface
  - AHB5 specification
  - Latest version of the MPU
- Offering optional
  - Micro Trace Buffer
  - Wakeup Interrupt Controller
  - fast I/O bus such as the Cortex-M0+
  - ETM (Embedded Trace Macro cell)
- Extends the number of maximum interrupts to 240
- Updated debug components
  - enhance debug operations
  - simplified usage

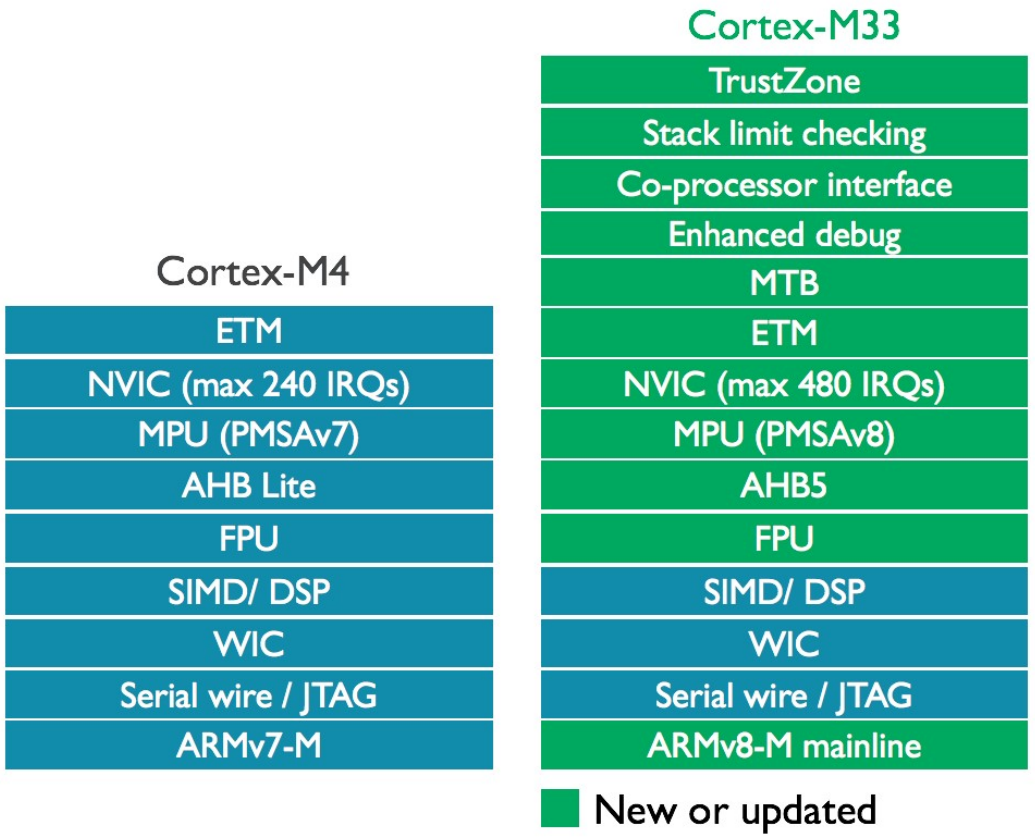


Armv8m baseline Cortex-M23

- Exclusive memory access instructions to simplify multi-core designs
- Instructions for divide operations to boost performance
- Stack limit checking in hardware when security is implemented
- TrustZone for software and hardware isolation

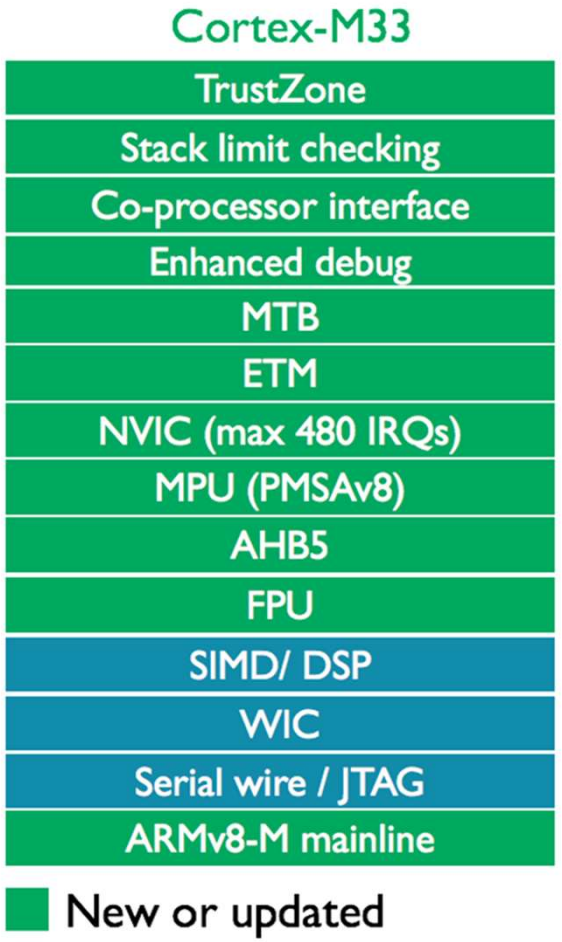


Armv8m mainline



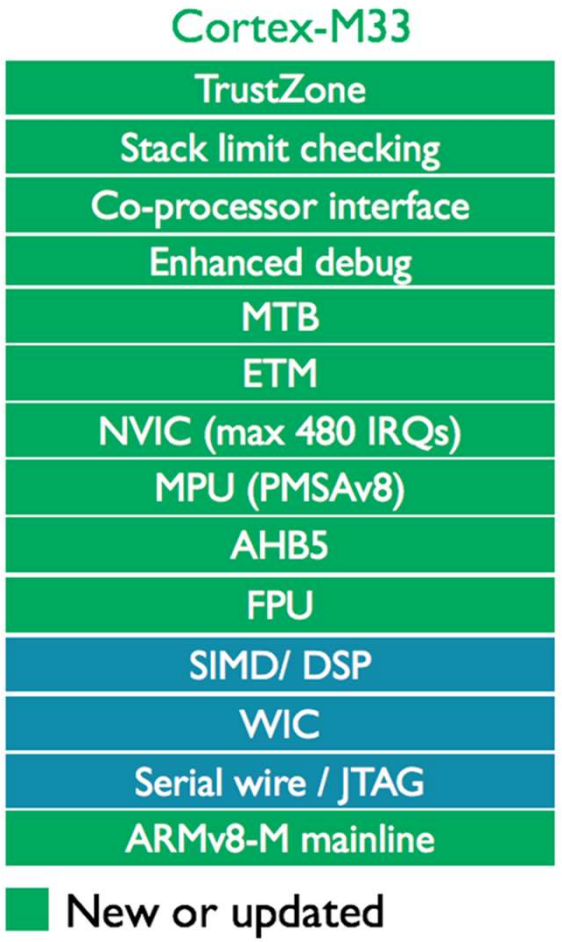
Armv8m mainline Cortex-M33

- Cortex-M33 implements the ARMv8-M architecture.
- Uses
  - Same debug
  - Same WIC functionality (Wakeup Interrupt Controller)
  - AHB5 specification
  - Latest version of the MPU
- Implementing
  - Same DSP/SIMD instructions as the Cortex-M4
  - Latest FPU specification which adds more instructions beyond what Cortex-M4 has
- Extends the number of maximum interrupts to 480



Armv8m mainline Cortex-M33

- Updated
  - ETM (Embedded Trace Macro cell)
    - A micro trace buffer as an option to trace into memory instead of out to the trace interface
  - Debug components
    - enhance debug operations
    - simplify usage
- Implements
  - Co-Processor Interface that supports up to 8 co-processors
  - Stack limit checking in hardware
  - TrustZone for software and hardware isolation



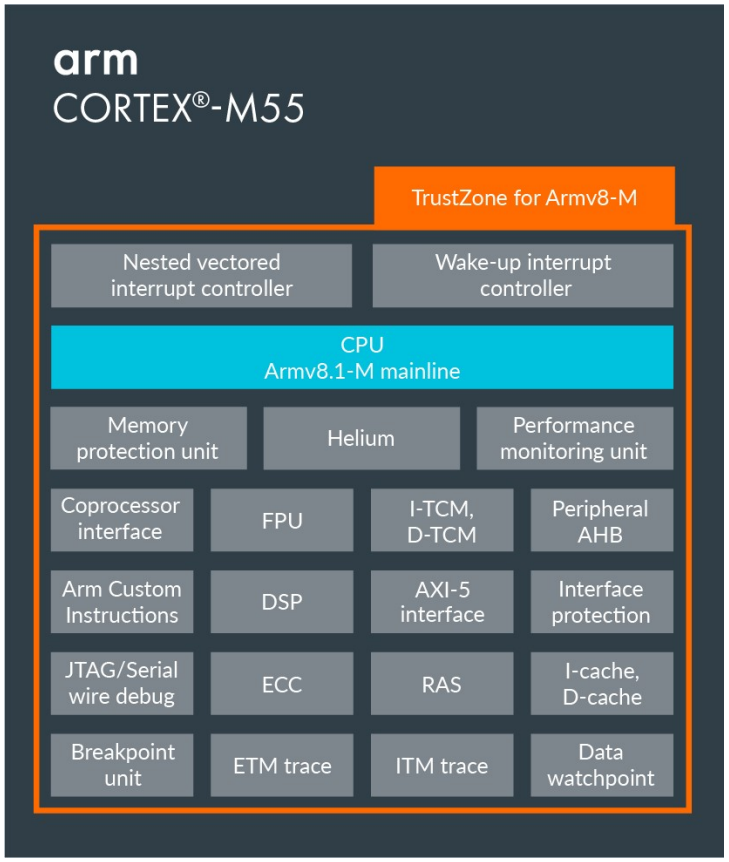
arm

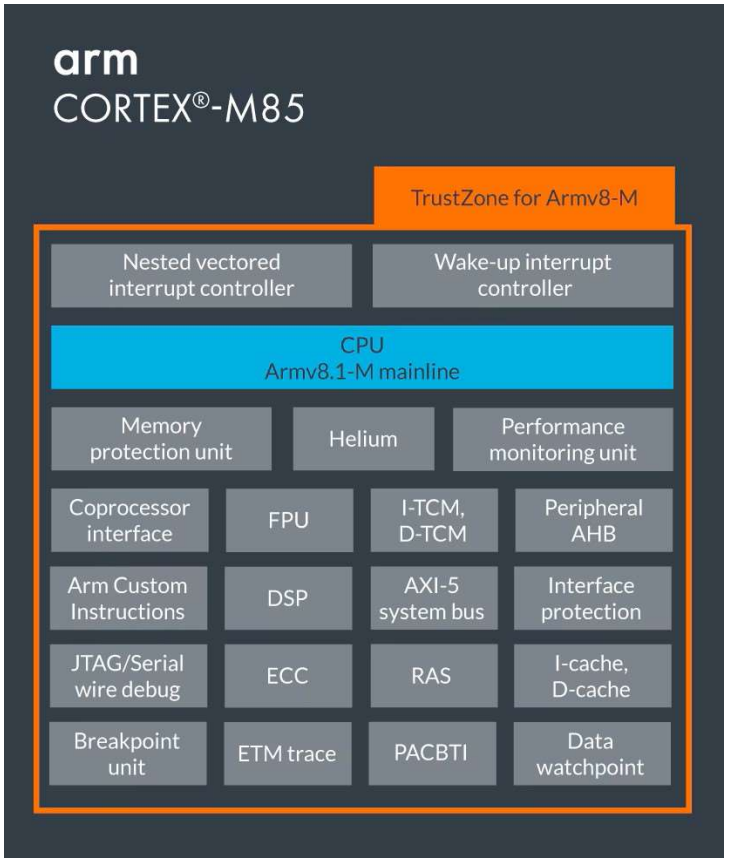
Armv8m differences

	Cortex-M23	Cortex-M33
Instruction set architecture	ARMv8-M baseline	ARMv8-M mainline
DMIPS/MHz	0.98*	1.50*
CoreMark®/ MHz	2.50*	3.86*
Bus interfaces	1xAHB5 + single cycle I/O	2xAHB5
Co-processor interface	No	Yes (option)
Number interrupts	1-240	1-480
Interrupt priorities	4	8-256
Interrupt latency	15 or 24 cycles	12 or 21cycles
DSP extension / SIMD / MAC	No	Yes (option)
SP floating point unit	No	Yes (option)
Breakpoints, watchpoints	0-4, 0-4 (option)	0-8, 0-4 (option)

arm

Armv8.1m – What has changed? What's new?



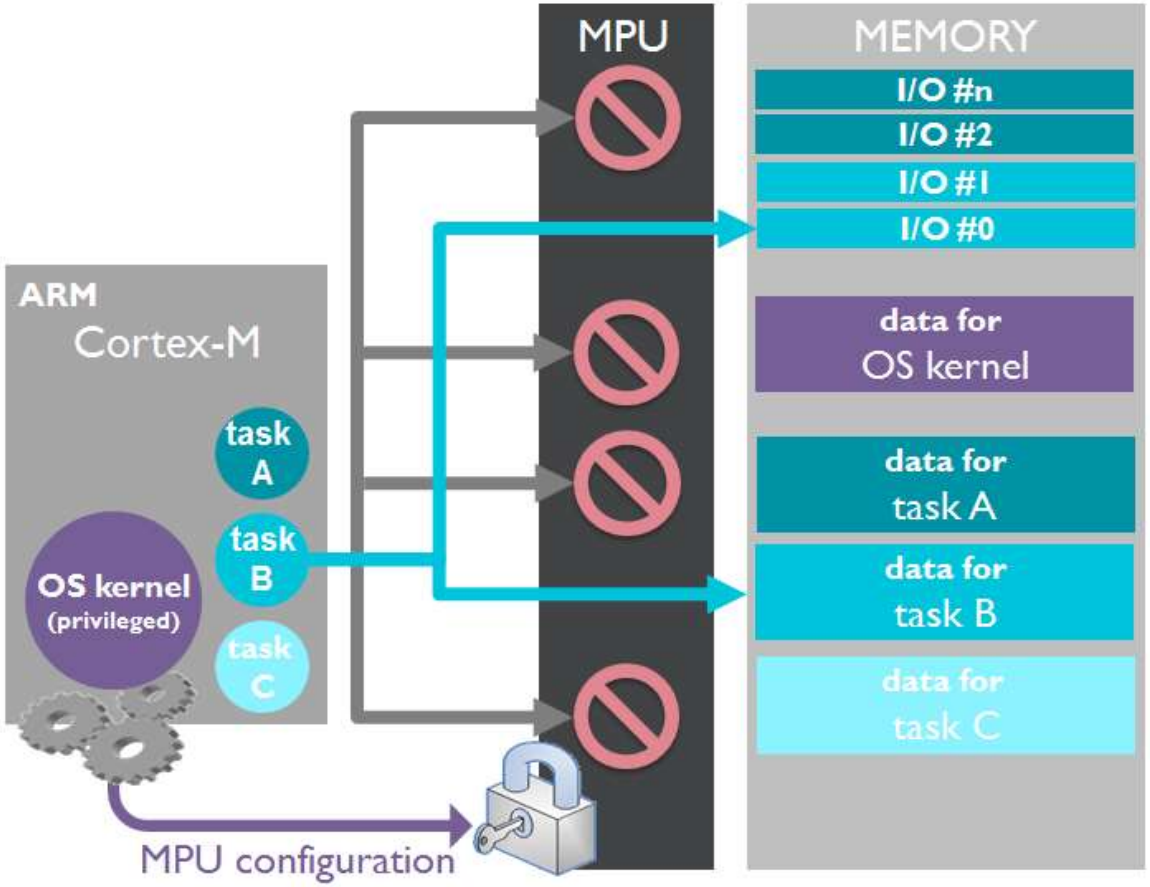


Armv8.1m – What has changed? What's new?

- Helium
  - New Vector Instruction Set extension
- Additional instruction set enhancements
  - Loops
  - Branches (Low Overhead Branch Extension)
  - Half precision floating-point support
  - TrustZone management for Floating Point Unit (FPU)
- New memory attribute in the Memory Protection Unit (MPU)
  - Privileged execute-never (PXN) attribute
- Enhancements in debug including
  - Performance Monitoring Unit (PMU)
  - Unprivileged Debug Extension
  - Additional debug support to focus on signal processing application developments
- Reliability, Availability and Serviceability (RAS) extension

MPU – Memory Protection Unit

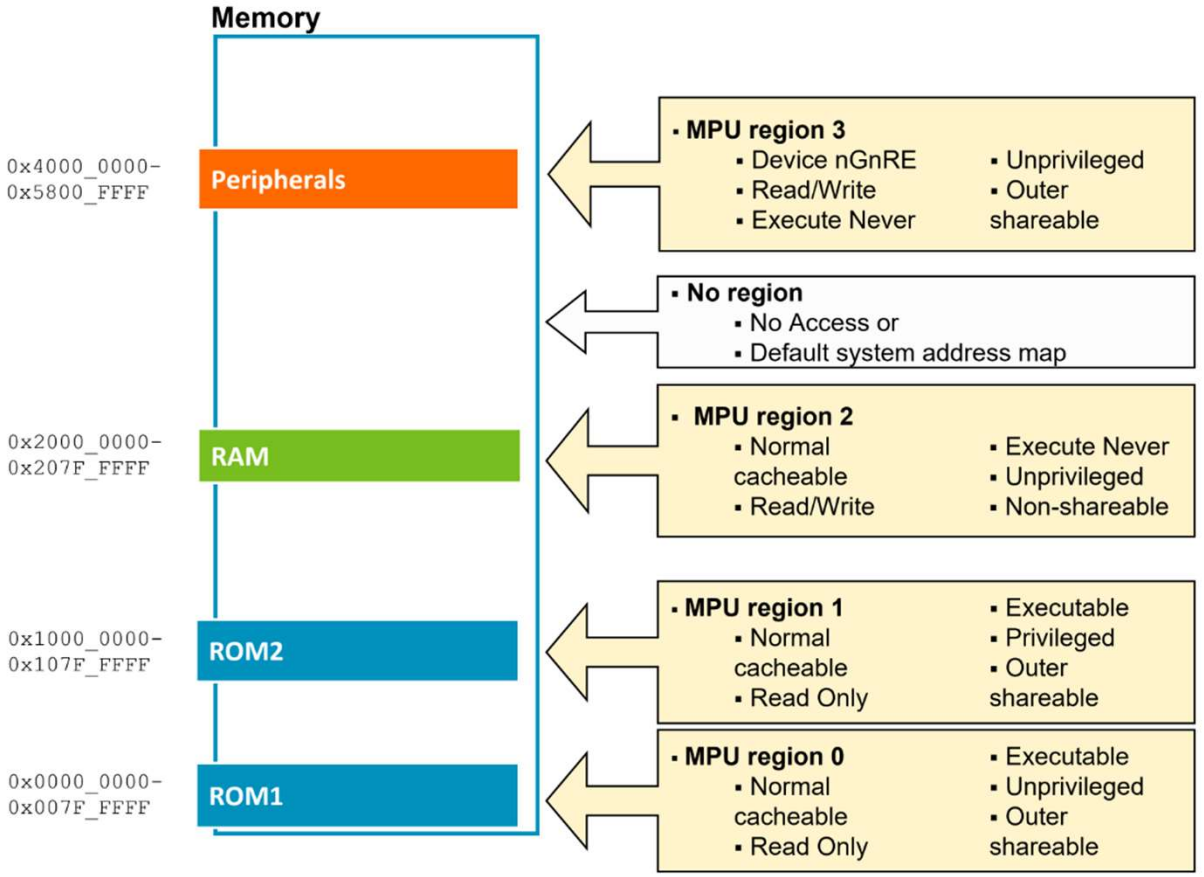
- Allows privileged software to define memory access permissions and memory attributes to different regions
- All memory access is monitored by the MPU, which can trigger a fault exception if unauthorized access is attempted
- Implements ARM Protected Memory System Architecture (PMSAv8)



arm

MPU – Memory Protection Unit

- MPU configuration
  - Fixed number of regions
  - For each region:
    - Base and limit address
    - Read-only/Read-write
    - Privileged/Unprivileged
    - Execution permission
    - Shareability
    - Cacheability (Normal memory)
    - Device attributes (peripherals)
  - Privileged software can use default memory map (PRIVDEFENA)

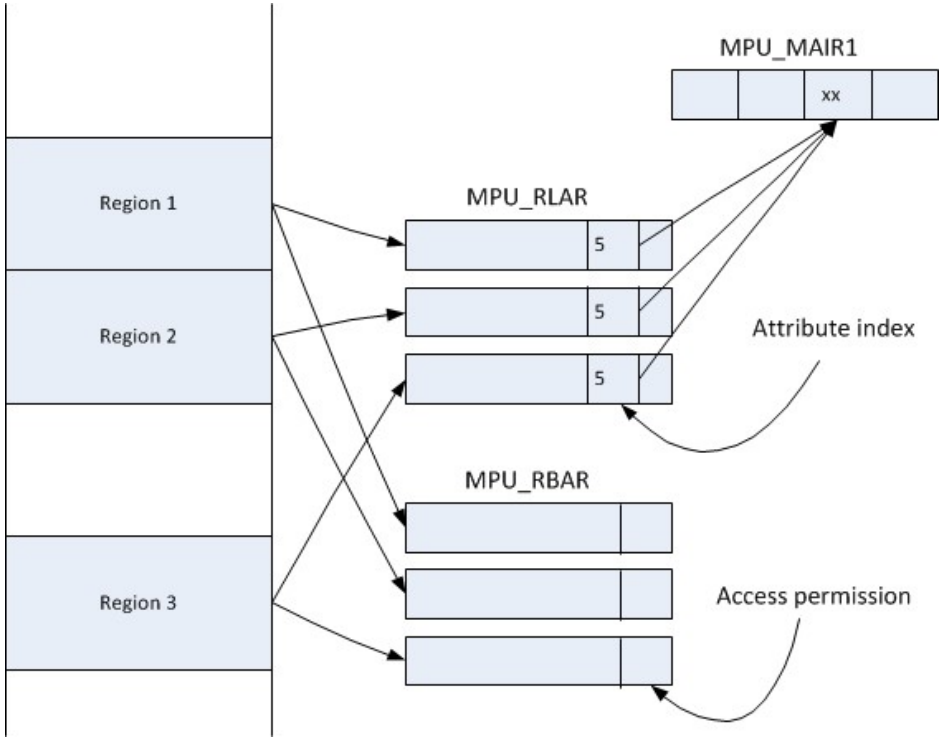


MPU – Memory Protection Unit

Armv6-m/Armv7-m MPU	Armv8-m MPU
Region must be aligned to an address which is a multiple of the region size, and that the region size must be a power of two.	The start and end address of a region only need to be aligned to a 32 byte boundary.
MPU regions can overlap. Higher region numbers have higher priority when MPU regions overlapped.	Regions are not allowed to overlap.
Memory attributes for each region are programmed in the corresponding MPU_RASR register.	Memory regions define memory attributes using an index into a set of memory attribute registers.
The concept of sub-regions is widely used within a single MPU region	There is no concept of sub-regions in PMSAv8. Because PMSAv8 gives more flexibility in region address configuration, there is no need to retain the sub-region concept used in Armv6-m and Armv7-m based processors.

MPU – Memory Protection Unit

- Armv8.1-m introduces Privileged execute-never (PXN) attribute
  - Ensures that privileged code cannot jump to unprivileged code and execute in privileged mode.

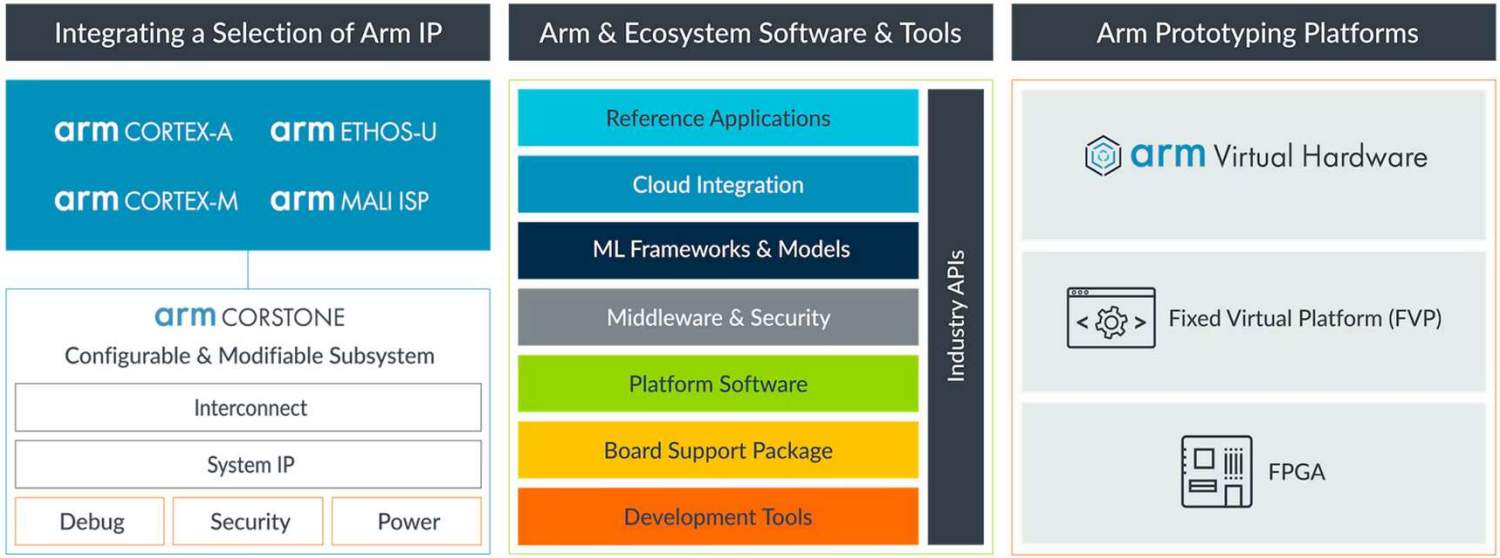


arm

Subsystems

- Is that all? What about system implementation?
  - What if DMA..? .. dual-core? Peripheral protection? etc.
- The answer is Compute Subsystems and the Corstone product line

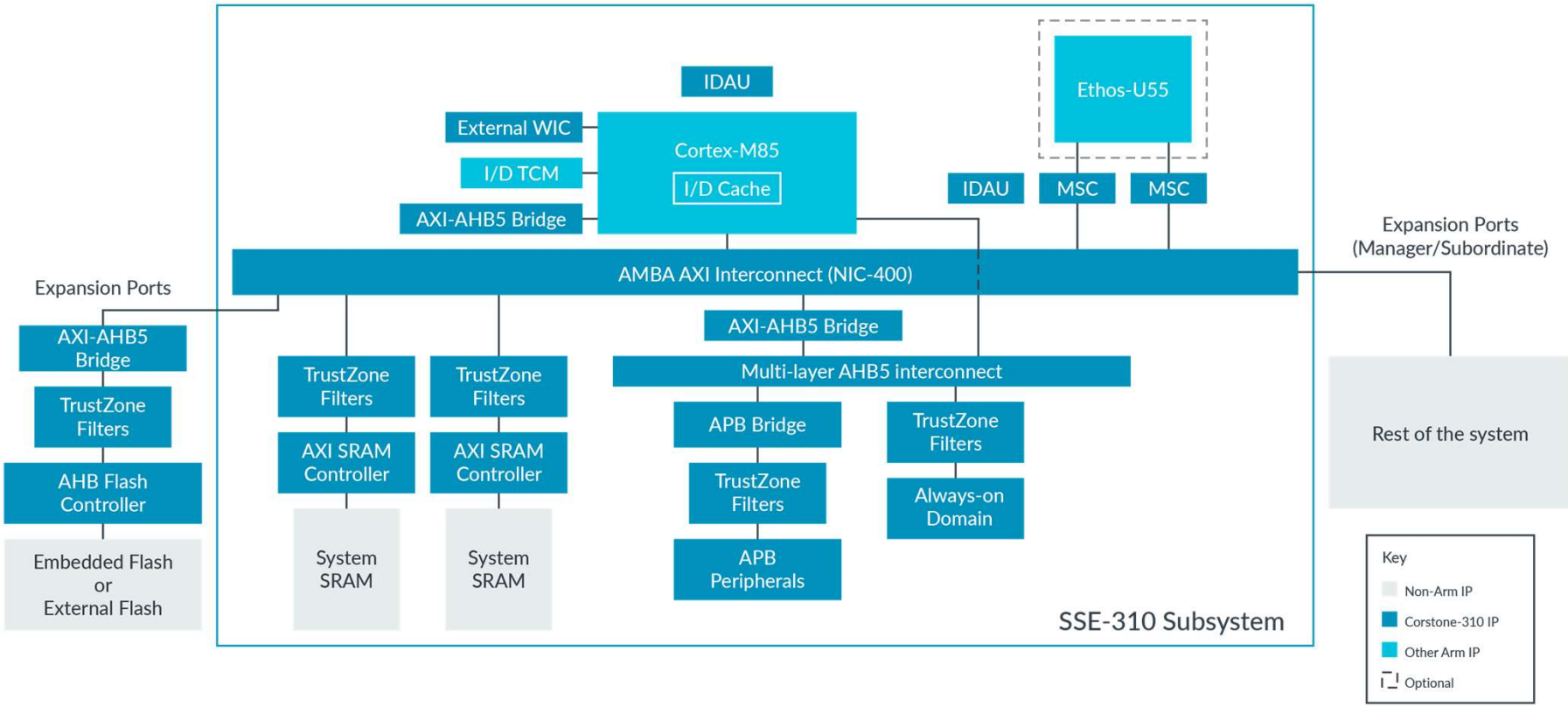
- Enabling Differentiation
  - Pre-integrated, pre-verified IP subsystem
- Faster Time to Market
  - Arm Virtual Hardware
- Security Built-In
  - PSA certification system



Subsystems

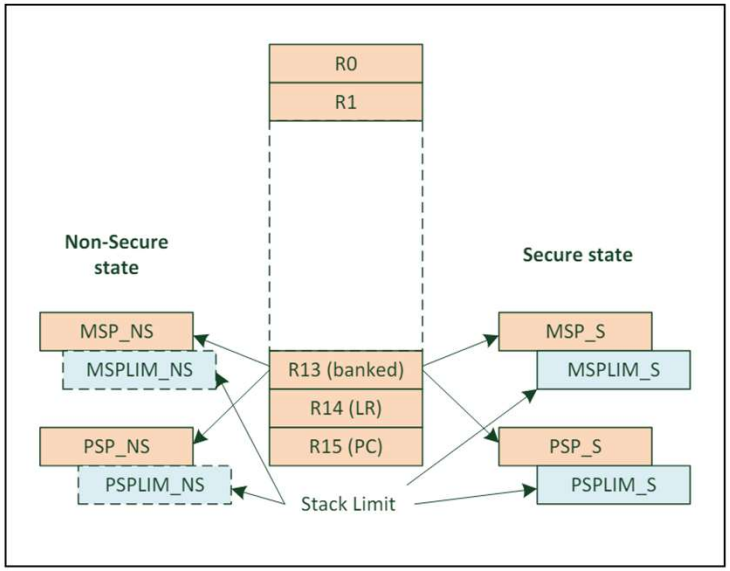
- Wide variety of IoT platforms
  - M-class baseline e.g. Corstone-102, Corstone-201
  - M-class mainline e.g. Corstone-300, Corstone-310
  - A+M: Corstone-1000
- Implementing Arm® Corstone™ Reference Systems Architecture Specification Ma1
- Arm Security IPs and more
- FVP (Fixed Virtual Platform) and FPGA support
- Reference systems with Open-Source Software support (TF-M, CMSIS, FreeRTOS, …)
- Different use-cases
- Arm IoT Reference Design Comparison Table

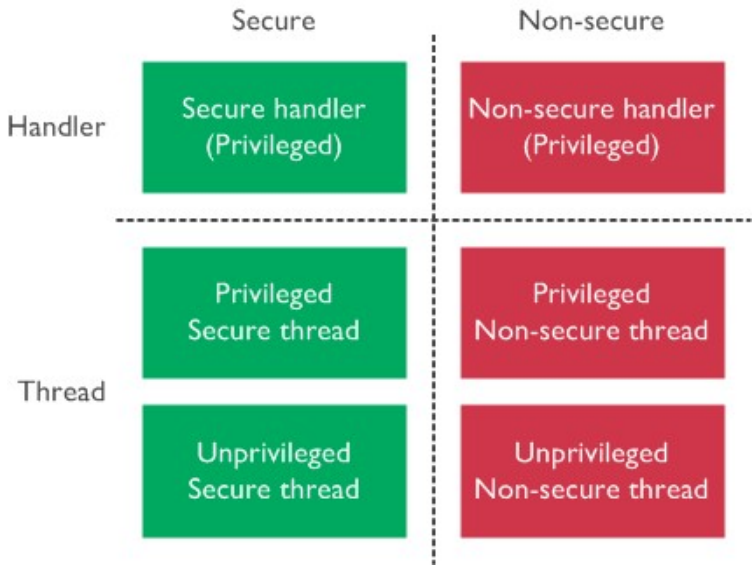
Subsystems



TrustZone

- TrustZone technology for Armv8-M/Armv8.1-M is an optional Security Extension
- The software running on the processor is divided into Normal Application(s) – NS side and secure firmware – S side, two separated applications, binaries
- Banked SysTick timer, MPU config, SCB and stack registers





TrustZone

- The advantages of having TrustZone
  - The attacker cannot access secure information in the secure memories.
  - The attacker cannot change the firmware because the firmware-update mechanism is protected.
  - TrustZone prevents the attacker from bypassing the product's life cycle management (debug access, downgrade)
  - A device vendor can include software libraries in the Secure world of a TrustZone enabled device without releasing the source code
- TrustZone does not prevent physical attacks
  - Fault injection attacks (glitches, power outage)
  - Side channel leakage (power, radiation, cache -> timing)
  - For additional physical protection check M35P

TrustZone

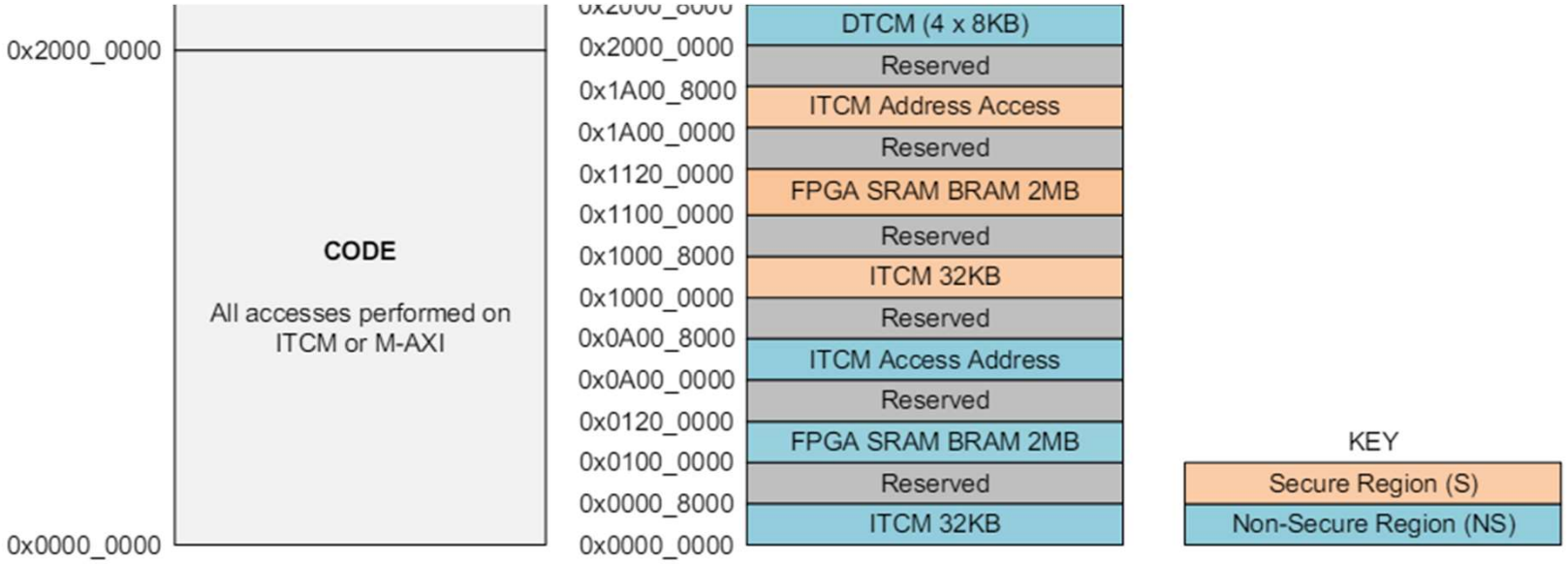
- In ARMv8-M, the memory space is partitioned into Secure and Non-Secure sections
- New IPs:
  - Master Security Controller (MSC)
  - Memory Protection Controller (MPC)
  - Peripheral Protection Controller (PPC)
  - Implementation Defined Attribution Unit (IDAU) + Security Attribution Unit (SAU)
- Three security attributes:
  - Secure
  - Non-secure
  - Secure and Non-secure callable

IDAU	SAU	Final Security
S	*X*	S
X	S	S
NS	S-NSC	S_NSC
NS	NS	NS
S-NSC	NS	S-NSC

arm

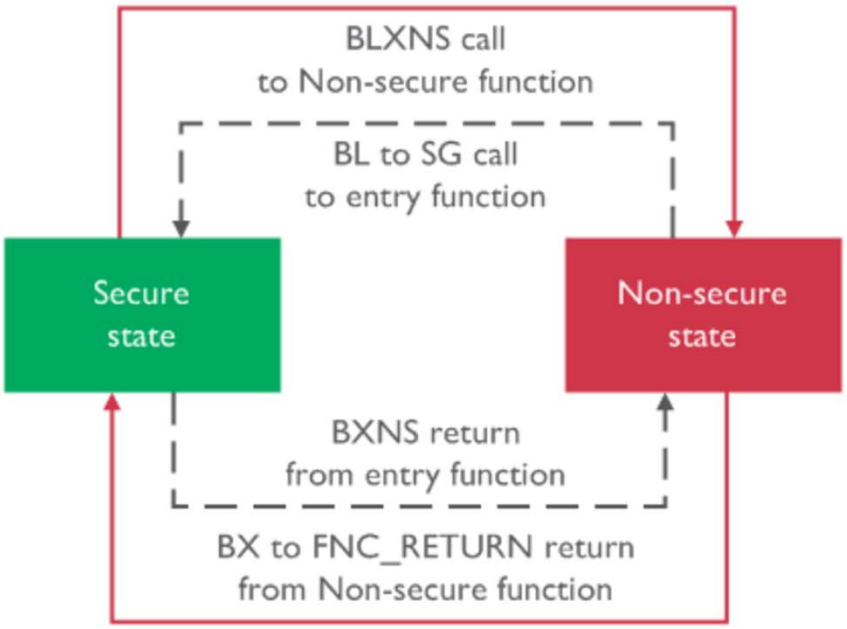
TrustZone

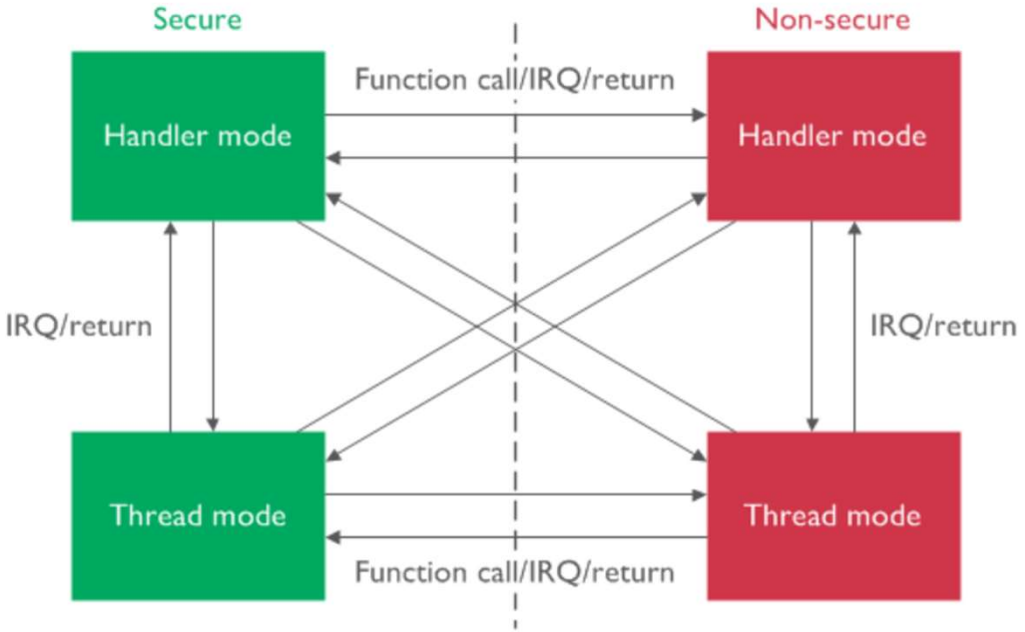
- Corstone-310 memory map (CODE/ROM part)
- Aliased regions
- Same applies for the peripherals



TrustZone

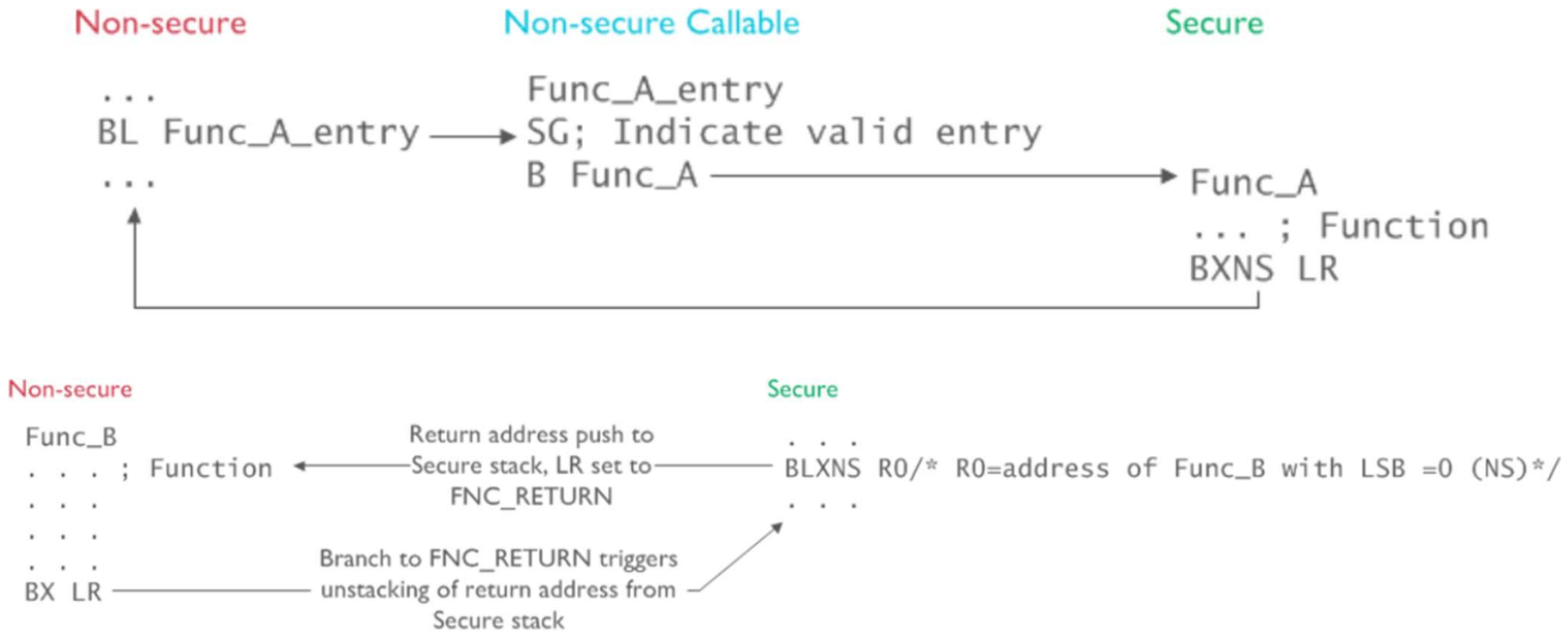
- Veneer functions for NS->S calls
- Register save/restore (normal vs. IRQ), Cortex-M Security Extensions (CMSE) C language extension





arm

TrustZone

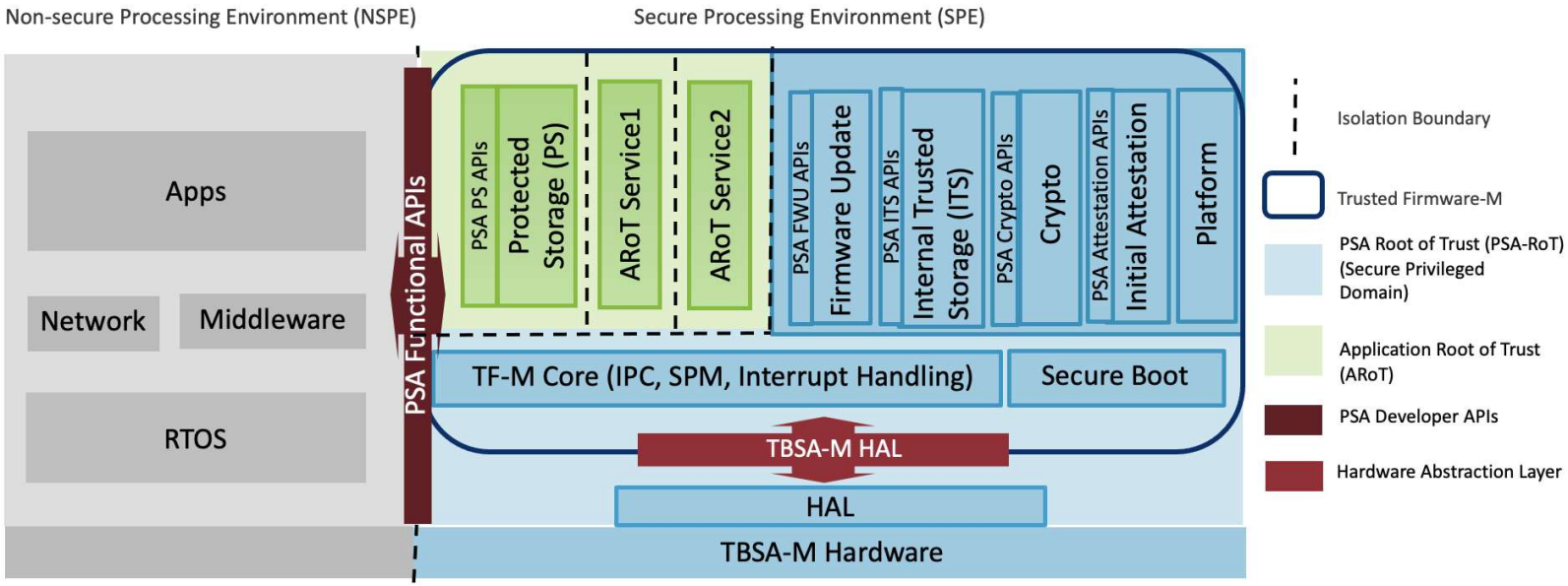


TrustZone – Use-cases

- IP protection
- Secure storage of critical information
- Root of trust implementation
- Sandboxing of certified software

Trusted Firmware M

- It is the platform security architecture reference implementation aligning with PSA Certified guidelines, enabling chips, Real Time Operating Systems and devices to become PSA Certified.
- Arm TrustZone or dual core architecture

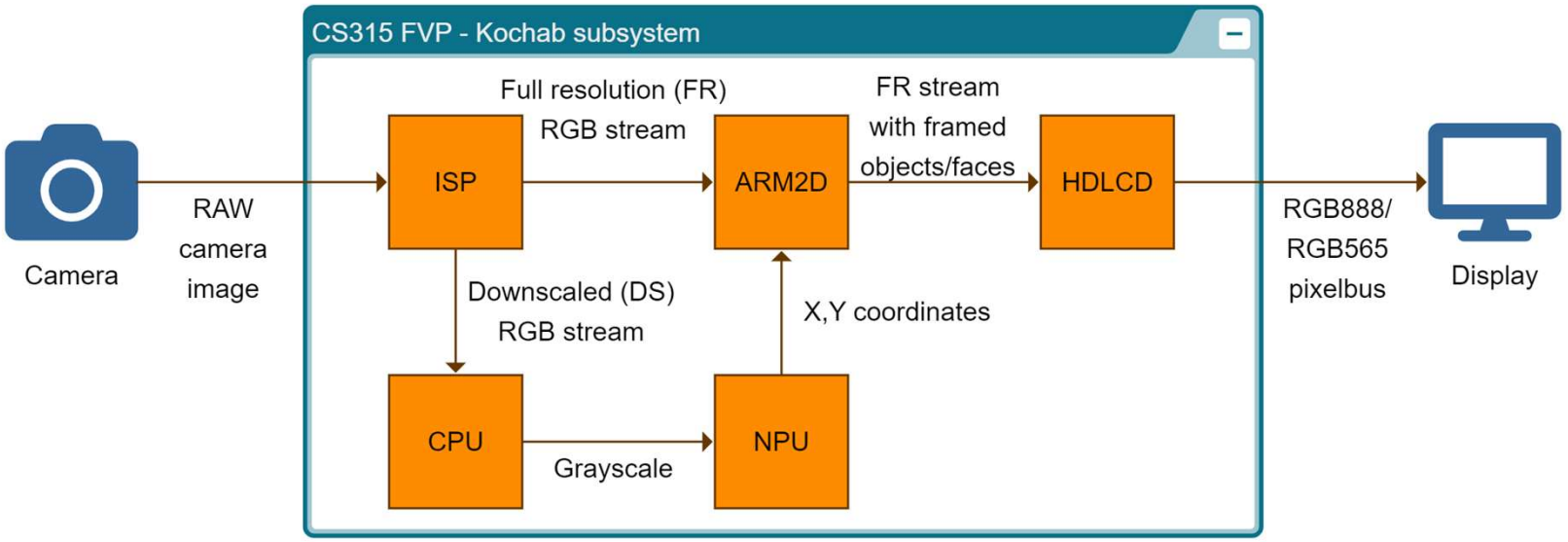


Trusted Firmware M

- It is like a library with secure services – NS side can use them
- You can add your own secure partition with your own secure services
- Provision your NS app’s credentials
- FWU support, anti-rollback protection, OTP counters
- S and NS image validation
- Crypto services
- Standardized PSA API
- Root-of-Trust

“Real” life example

- https://github.com/FreeRTOS/iot-reference-arm-corstone3xx
- Object detection example
  - Face recognition on a 1080p30Hz video stream
  - Corstone-315 (M85, U65, DMA350, Mali-C55 ISP, HDLCD, TF-M)



arm

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