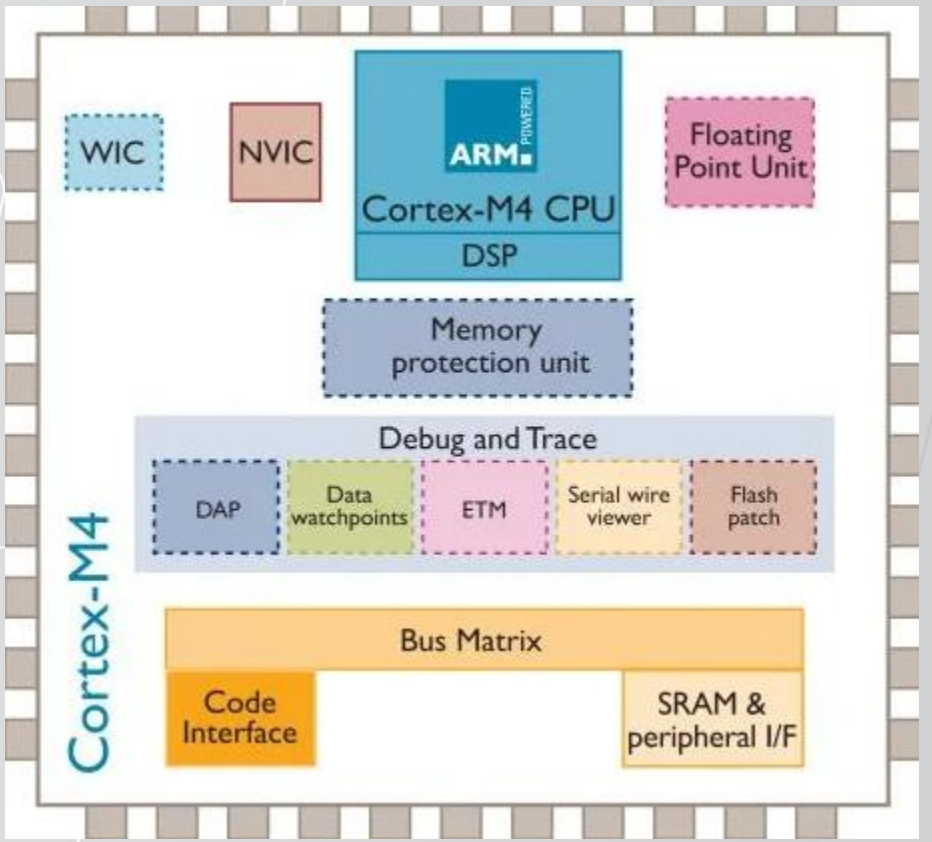
ARM Architecture

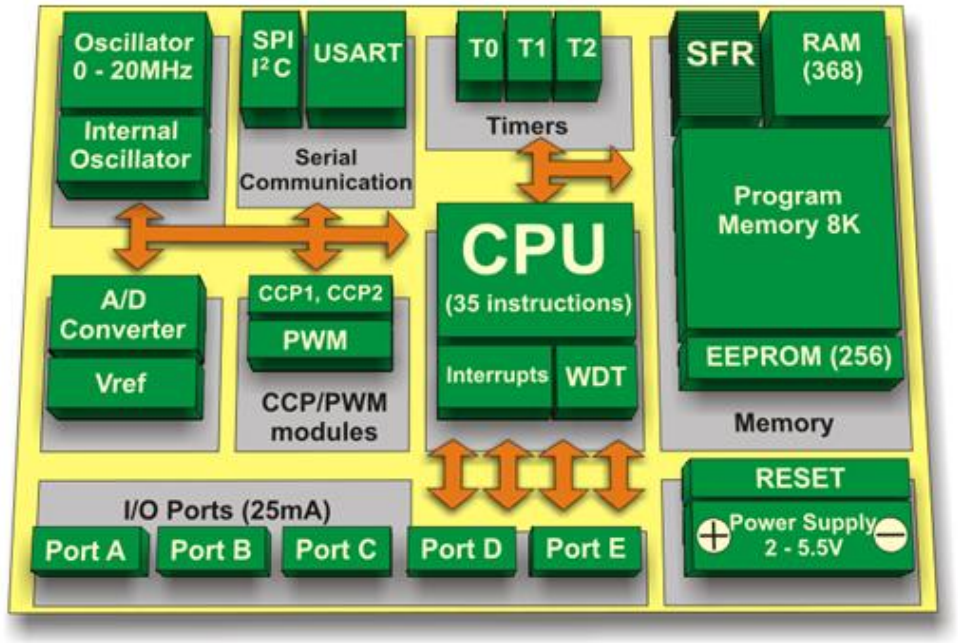


24 Aug 2024

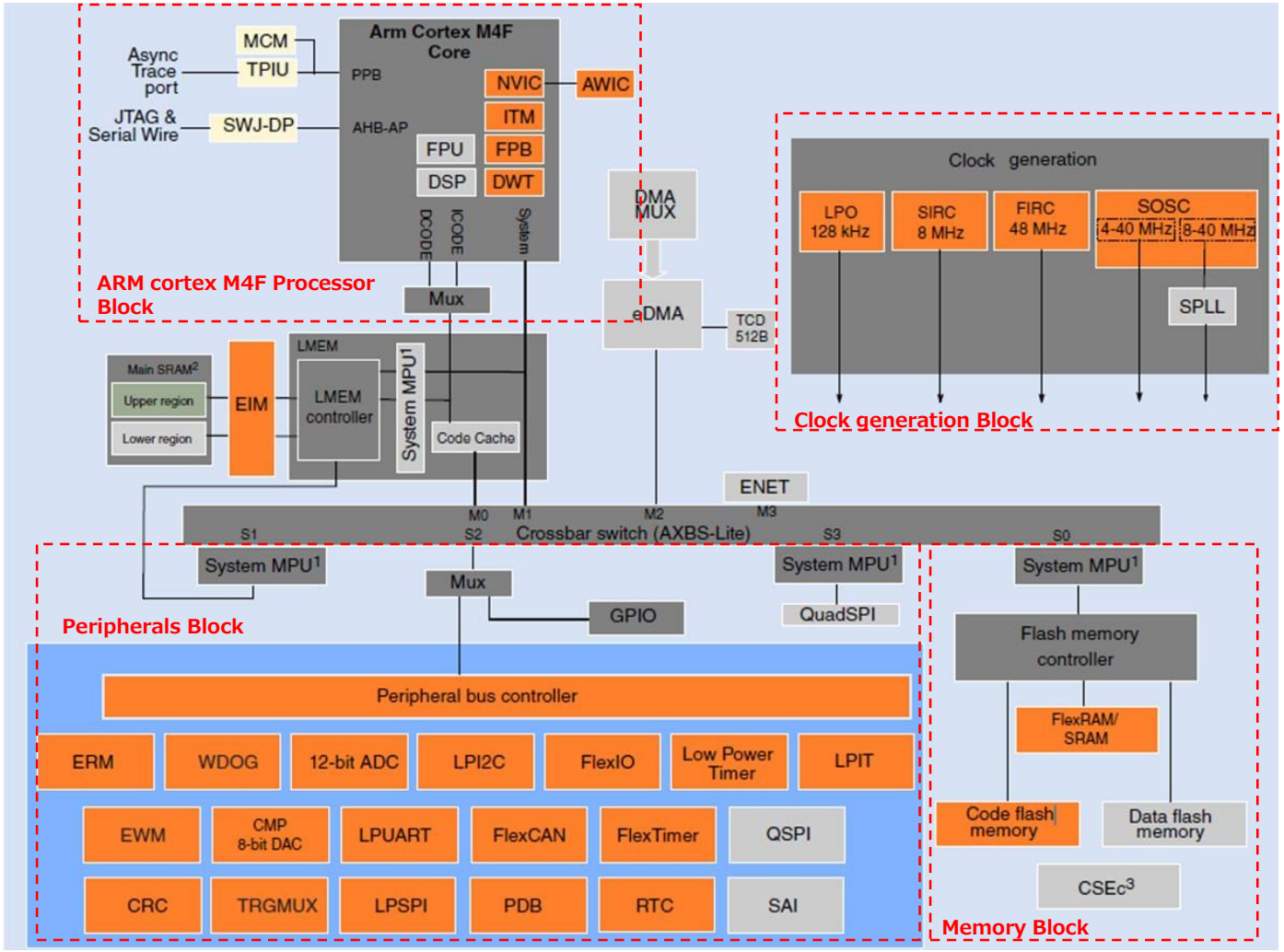
OUTLINE

1. Microcontroller
2. ARM Limited
3. List of ARM processors
4. ARM Cortex-M3/M4/M4F/M7
5. ARM Cortex-M4
    - 5.1. Block diagram
    - 5.2. Processor
    - 5.3. Floating Point Unit (FPU)
    - 5.4. NVIC
    - 5.5. Memory Protection Unit (MPU)
    - 5.6. Debug and Trace
    - 5.7. Bus interfaces
    - 5.8. Programmers model
    - 5.9. Memory model
    - 5.10. ARM register set
    - 5.11. Bit-banding
    - 5.12. Processor core register summary
    - 5.13. Exceptions

1. Microcontroller



❖ **Structure of Microcontroller:**
- + CPU
- + Memory
- + Peripheral(PORT, TIMER,…)
- + Clock

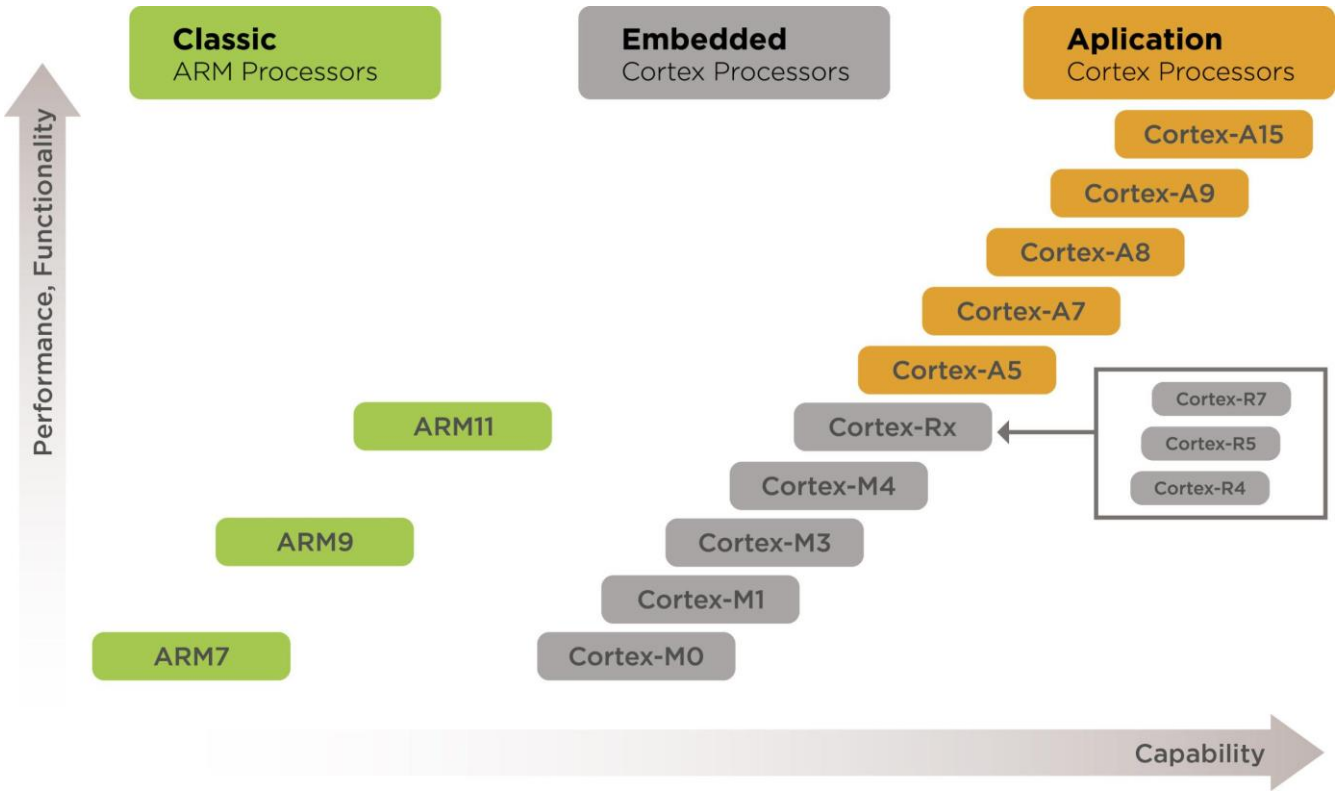


S32K14X BLOCK DIAGRAM

2. ARM Limited

- **ARM** founded in November 1990
    - **A**dvanced **R**ISC **M**achines
- Company headquarters in Cambridge, UK
    - Processor design centers in Cambridge, Austin, and Sophia Antipolis
    - Sales, support, and engineering offices all over the world
- Best known for its range of RISC processor cores designs
    - Other products — fabric IP, software tools, models, cell libraries - to help partners develop and ship ARM-based SOCs
- ARM does **not** manufacture silicon

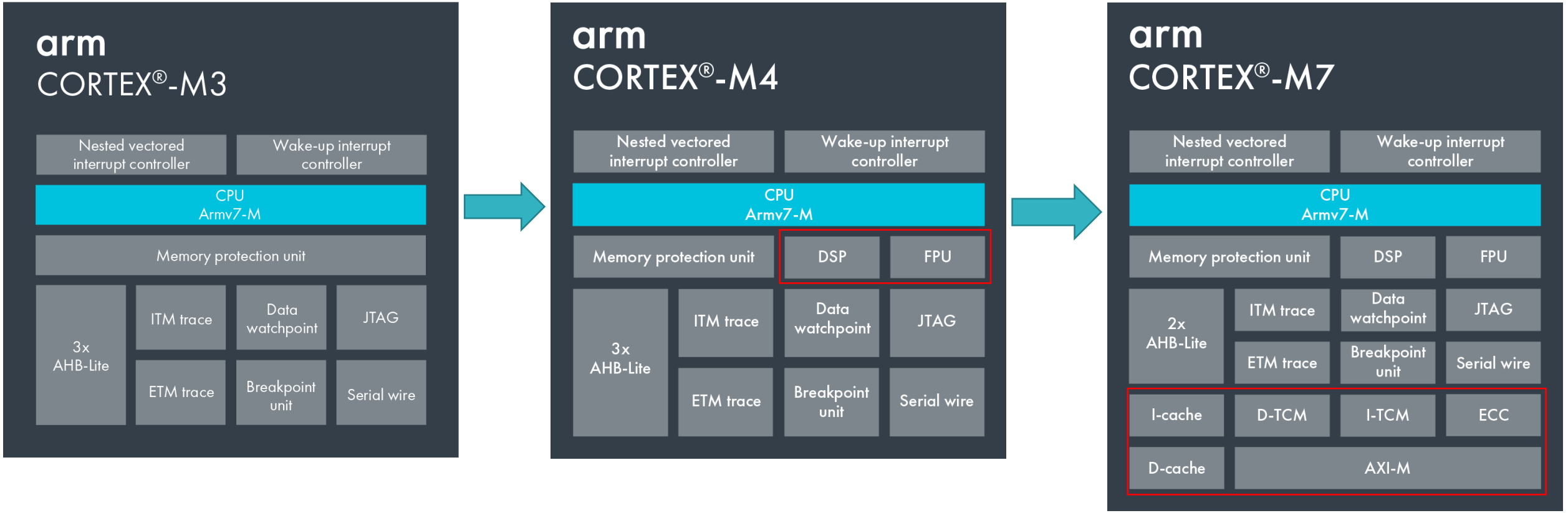
3. List of ARM processors



Cortex-A: Applications
Cortex-R: Real-time application
Cortex-M: Microcontrollers

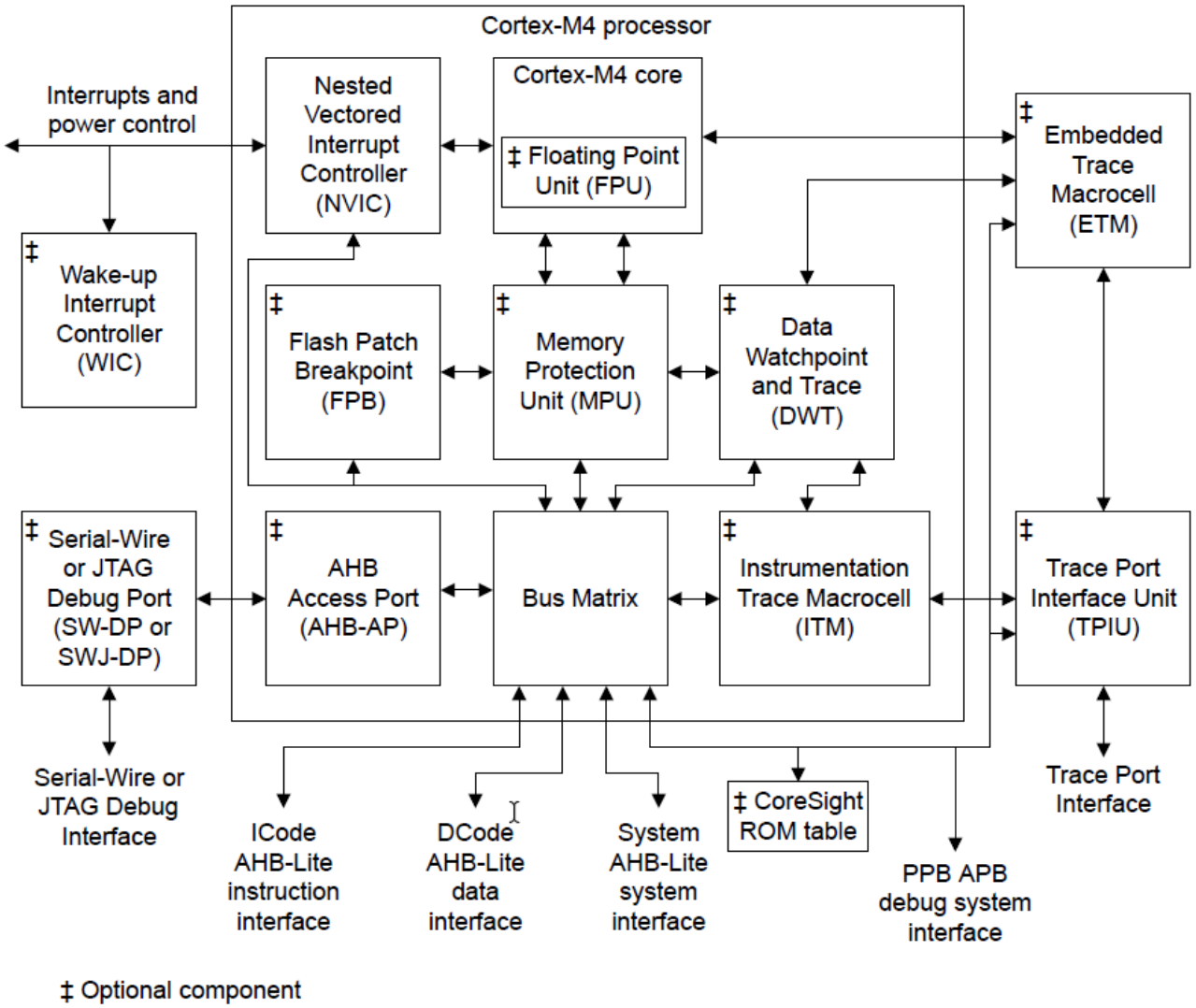
ARMv1 → ARMv9 architecture.
ARMv6-M: ARM Cortex-M0, ARM Cortex-M0+, ARM Cortex-M1
ARMv7-M: ARM Cortex-M3
ARMv7E-M: **ARM Cortex-M4**, ARM Cortex-M7
ARMv8-M: ARM Cortex-M23, ARM Cortex-M33
ARMv8.1-M: ARM Cortex-M55, ARM Cortex-M85

4. ARM Cortex-M3/M4/M4F/M7

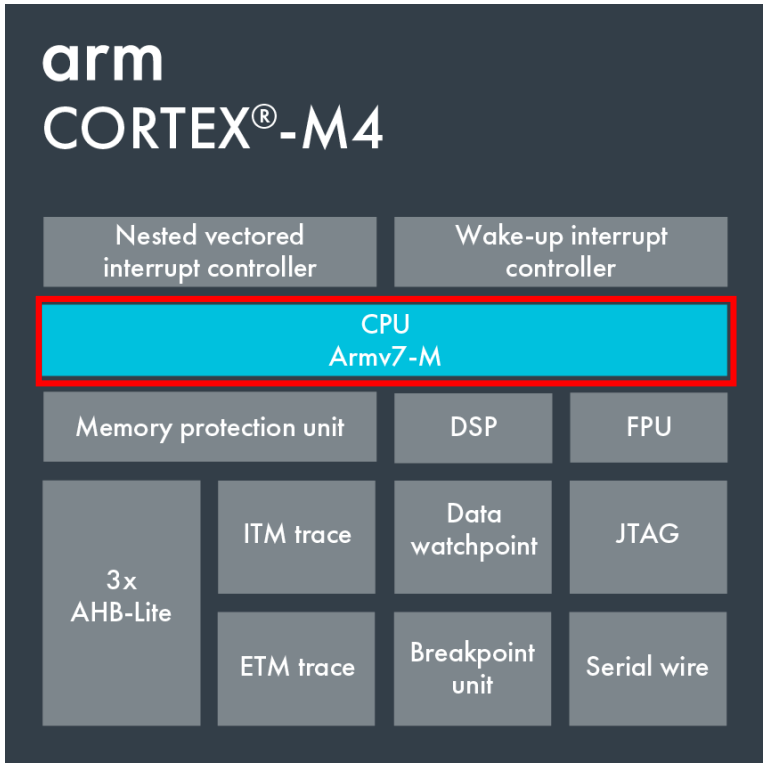


Cortex-M4 is a Cortex-M3 plus DSP instructions, and optional floating-point unit (FPU). A core with an FPU is known as Cortex-M4F.

5.1. Block diagram



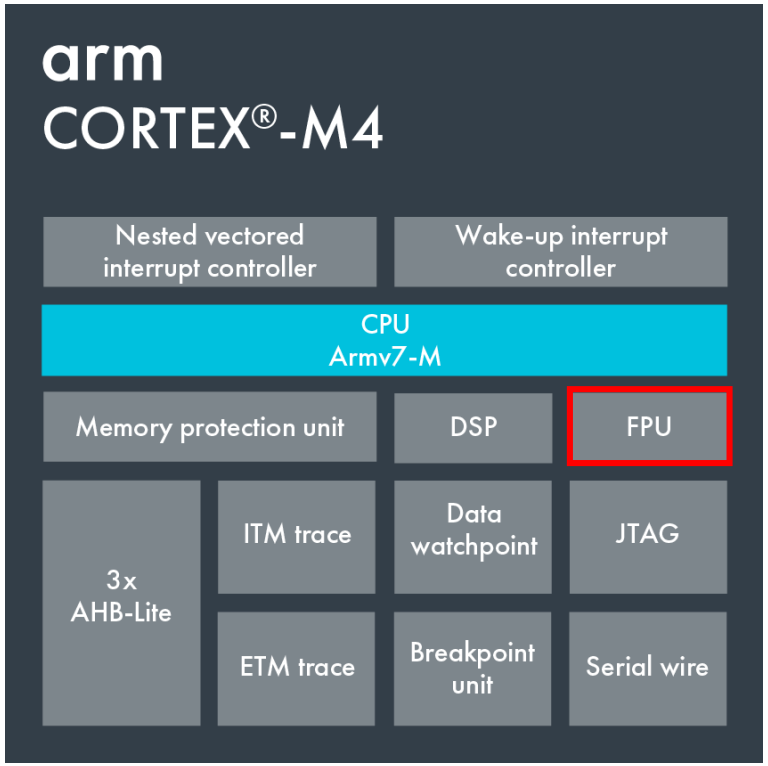
5.2. Processor



The Cortex-M4 processor features a **low gate count processor** core, with **low latency interrupt processing** that has:

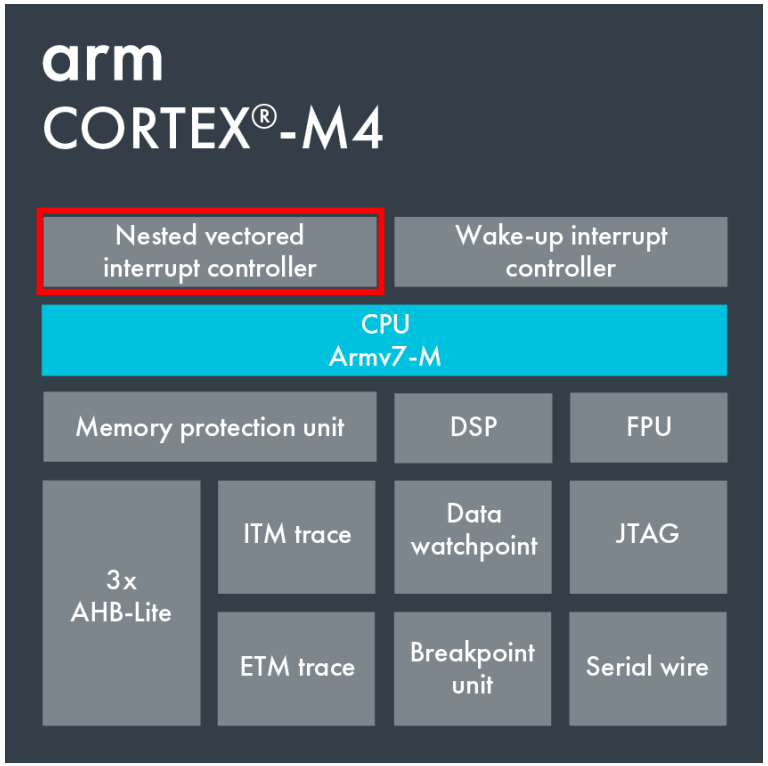
- A subset of the **Thumb** instruction set, defined in the Armv7-M architecture
- Banked Stack Pointer (SP)
- Hardware divide instructions, SDIV and UDIV
- **Handler** and **Thread** modes
- **Thumb** and **Debug** states
- Support for interruptible-continued instructions LDM, STM, PUSH, and POP for low interrupt latency
- Automatic processor state saving and restoration for low latency Interrupt Service
- Routine (**ISR**) entry and exit

5.3. Floating Point Unit (FPU)



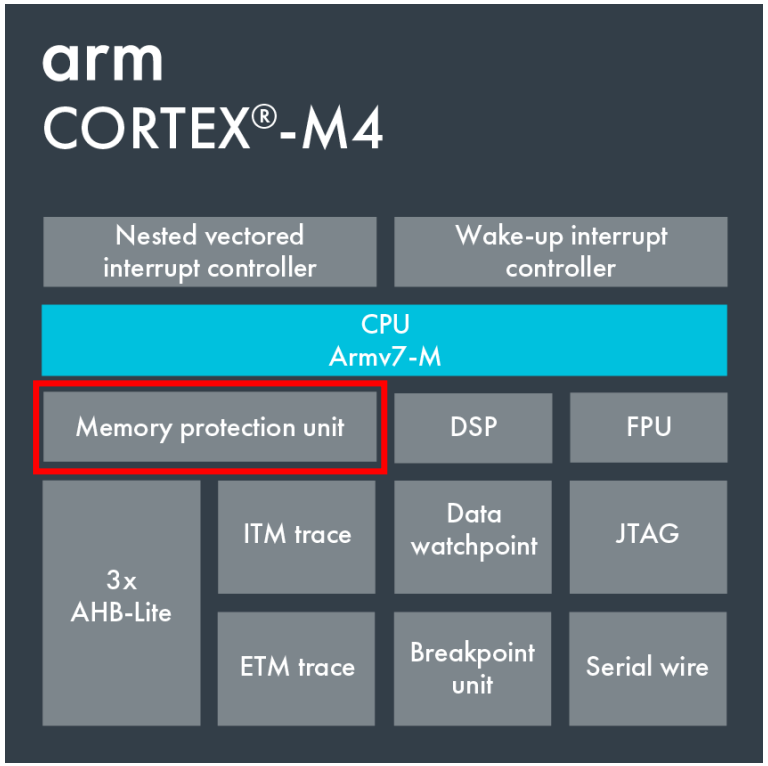
- 32-bit instructions for single-precision (C float) data-processing operations
- Combined Multiply and Accumulate instructions for increased precision (Fused MAC)
- Hardware support for conversion, addition, subtraction, multiplication with optional accumulate, division, and square root
- Hardware support for denormals and all IEEE rounding modes
- 32 dedicated 32-bit single precision registers, also addressable as 16
- Double-word registers
- Decoupled three-stage pipeline

5.4. NVIC



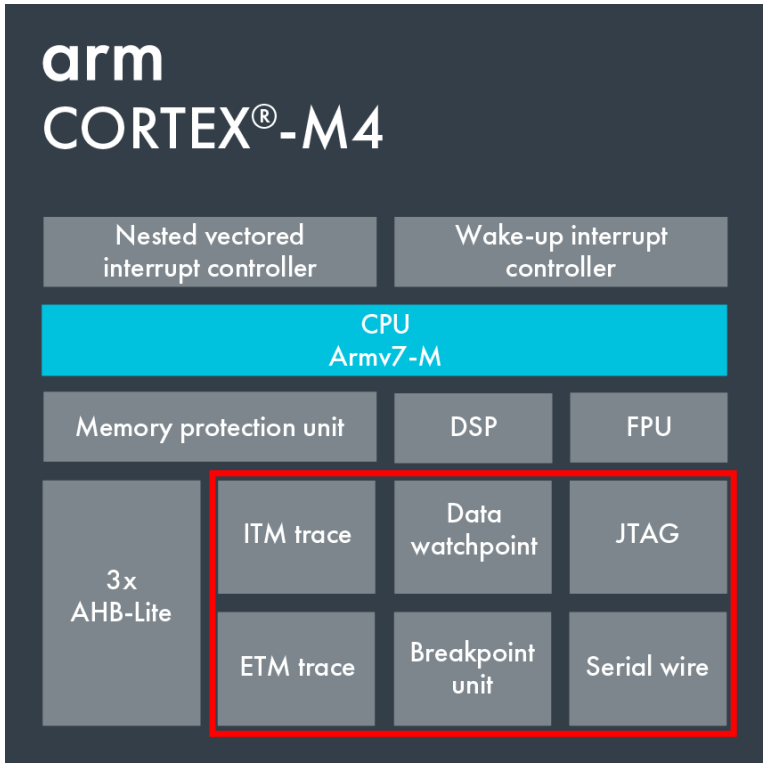
- **Nested Vectored Interrupt Controller** (NVIC) closely integrated with the processor core to achieve low latency interrupt processing. Features include:
  - External interrupts, configurable from 1 to 240
  - Bits of priority, configurable from 3 to 8
  - Dynamic reprioritization of interrupts
  - Priority grouping
    - This enables selection of preempting interrupt levels and non preempting interrupt levels.
  - Support for **tail-chaining** and **late arrival** of interrupts
    - This enables back-to-back interrupt processing without the overhead of state saving and restoration between interrupts.
  - Processor state automatically saved on interrupt entry, and restored on interrupt exit, with no instruction overhead
  - Optional Wake-up Interrupt Controller (WIC), providing ultra-low power sleep mode support

5.5. Memory Protection Unit (MPU)



- The MPU is a programmable device that can be used to define memory access permissions (e.g., privileged access only or full access) and memory attributes (e.g., bufferable, cacheable) for different memory regions.
- The MPU on CortexM3 and Cortex-M4 processors can support up to eight programmable memory regions, each with their own programmable starting addresses, sizes, and settings.
- The configuration can be used for functions like:
  - Setting a RAM/SRAM region to be read-only to protect important data from accidental corruption.
  - Making a portion of RAM/SRAM space at the bottom of the stack inaccessible to detect stack overflow.
  - Setting a RAM/SRAM region to be XN to prevent code injection attacks.

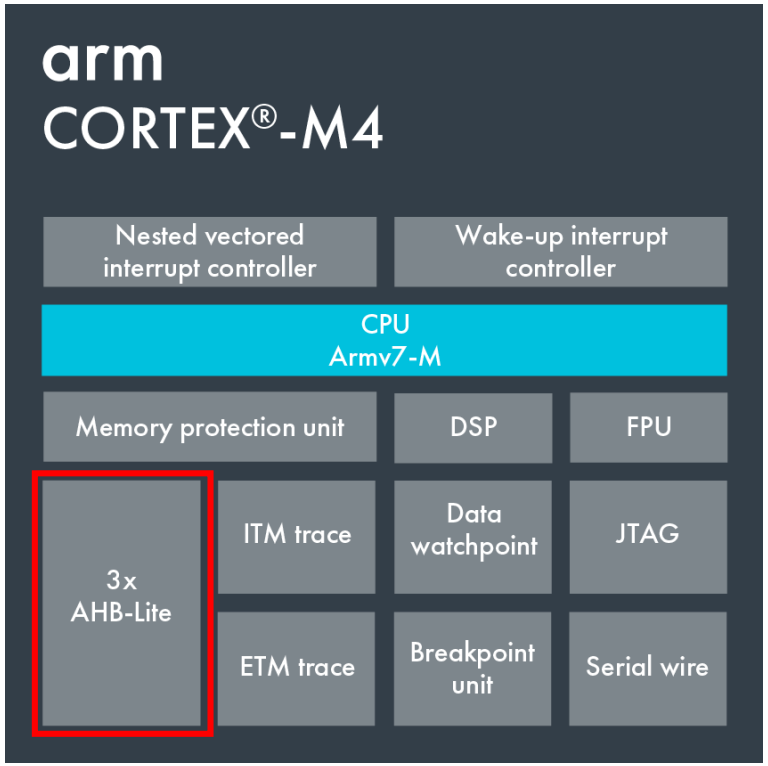
5.6. Debug and Trace



Low-cost debug solution that features:

- Debug access to all memory and registers in the system, including access to memory mapped devices, access to internal core registers when the core is halted, and access to debug control registers even while SYSRESETn is asserted
- Serial Wire Debug Port (SW-DP) or Serial Wire JTAG Debug Port (SWJ-DP) debug access, or both
- Optional Flash Patch and Breakpoint (FPB) unit for implementing breakpoints and code patches
- Optional Data Watchpoint and Trace (DWT) unit for implementing watchpoints, data tracing, and system profiling
- Optional Instrumentation Trace Macrocell (ITM) for support of printf style debugging
- Optional Trace Port Interface Unit (TPIU) for bridging to a Trace Port Analyzer (TPA), including Single Wire Output (SWO) mode
- Optional ETM for instruction trace

5.7. Bus interfaces



- Three *Advanced High-performance Bus-Lite* (AHB-Lite) interfaces: **ICode, DCode, and System.**
- Bus interfaces.
- *Private Peripheral Bus* (PPB) based on *Advanced Peripheral Bus* (APB) interface.
- Bit-band support that includes atomic bit-band write and read operations.
- Memory access alignment.
- Write buffer for buffering of write data.
- Exclusive access transfers for multiprocessor systems.

5.8. Programmers model

Operating mode:

1. **Thread mode:** Used to execute application software. The processor enters Thread mode when it comes out of reset.
2. **Handler mode**: Used to handle exceptions. The processor returns to Thread mode when it has finished all exception processing.

Operating states:

1. **Thumb state:** This is normal execution running 16-bit and 32-bit halfword aligned Thumb instructions.
2. **Debug State:** This is the state when the processor is in halting debug.

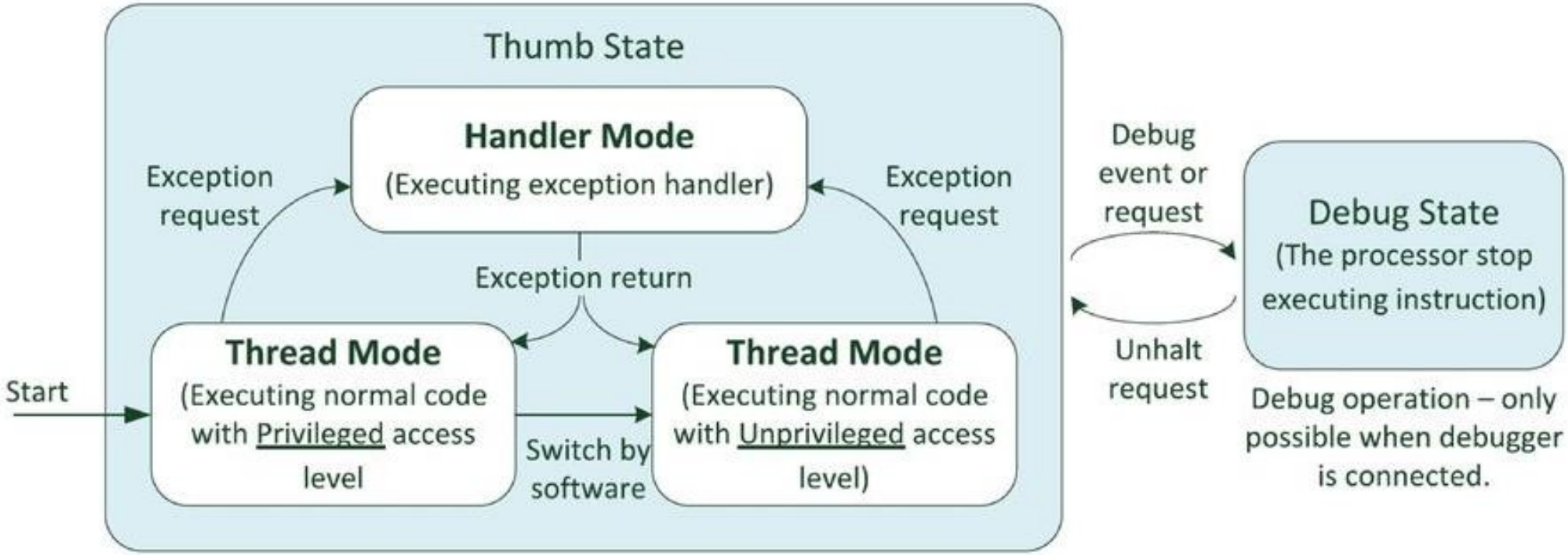
Privilege levels:

1. **Unprivileged:** The software:
   - has limited access to the MSR and MRS instructions, and cannot use the CPS instruction
   - cannot access the system timer, NVIC, or system control block
   - might have restricted access to memory or peripherals.
2. **Privileged:** The software can use all the instructions and has access to all resources.

Handler mode is always privileged. Thread mode can be privileged or unprivileged.

5.8. Programmers model

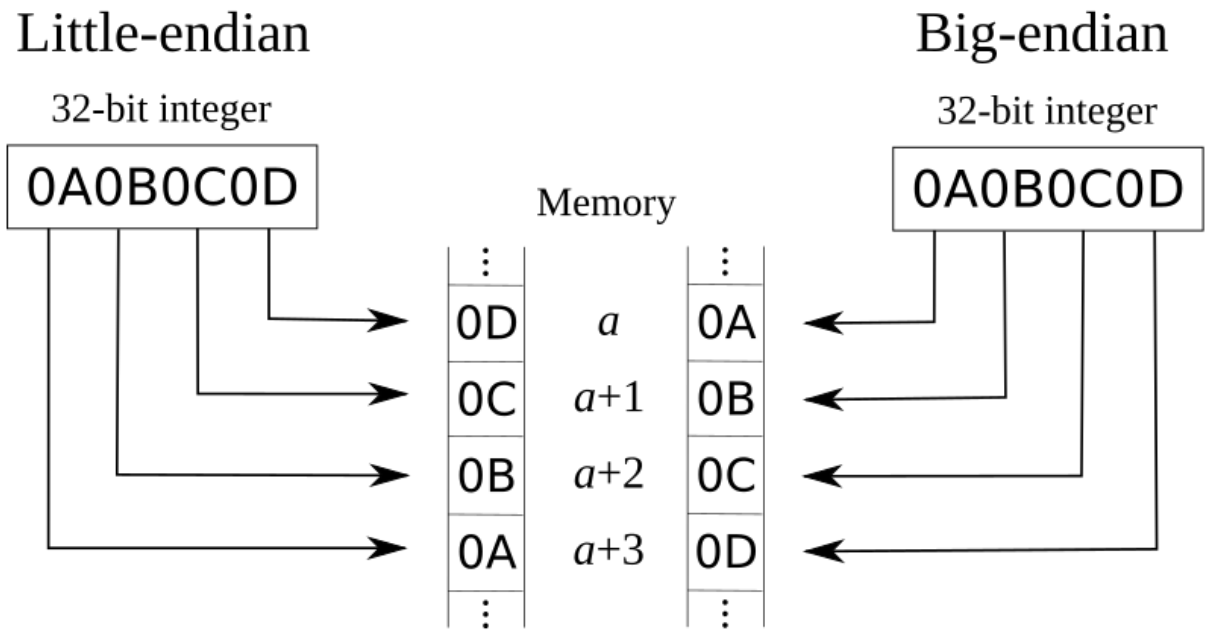
Operating modes and States:



5.8. Programmers model

Data types:

- 32-bit words
- 16-bit half-words
- 8-bit bytes



Endianness:

The order in which bytes are arranged in memory

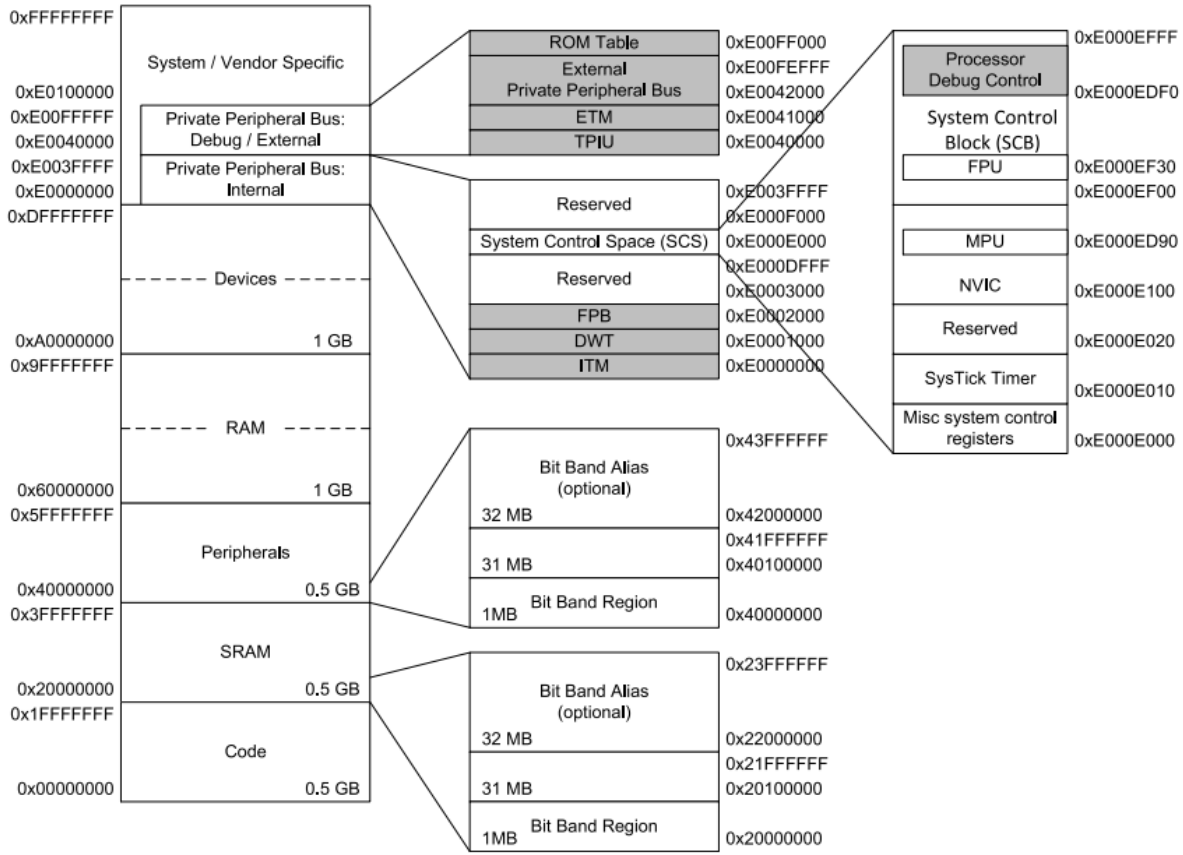
- **Big-endian (BE):** the most significant byte **(MSB)** is stored at the lowest memory address
- **Little-endian (LE):** the least significant byte **(LSB)** is stored at the lowest memory address.

❖ Instruction memory and *Private Peripheral Bus* (PPB) accesses are always performed as **little-endian.**

5.9. Memory model

Memory map – Regions:

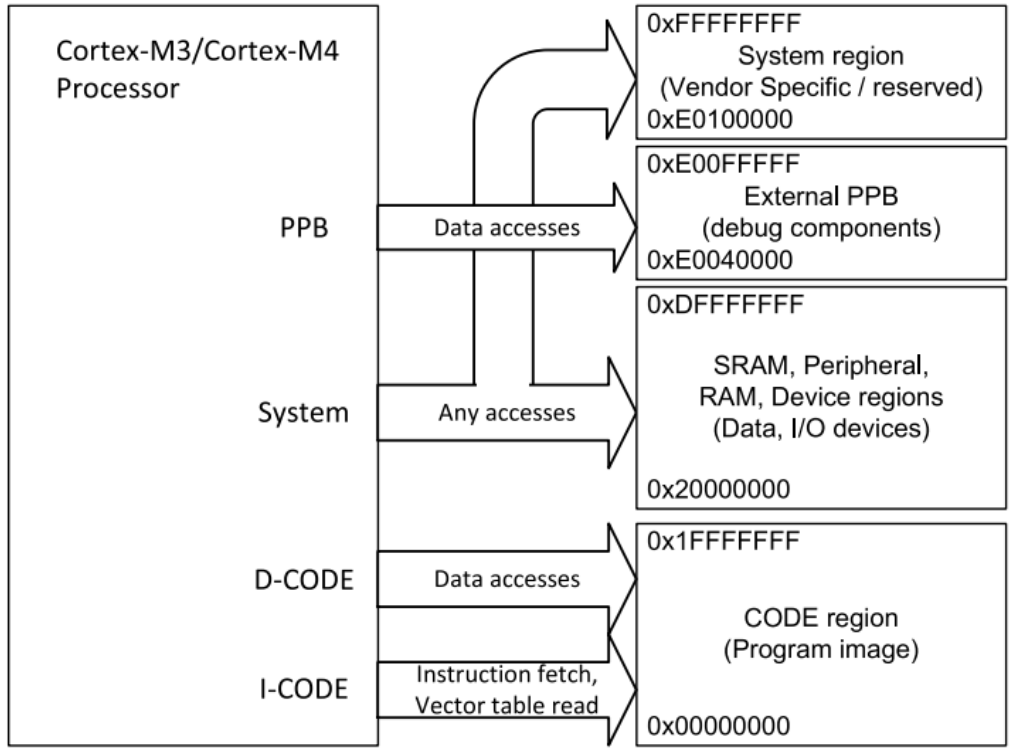
- Up to 4GB of addressable memory



Memory Map	Region
Code	Instruction fetches are performed over the ICode bus. Data accesses are performed over the DCode bus.
SRAM	Instruction fetches and data accesses are performed over the system bus.
SRAM bit-band	Alias region. Data accesses are aliases. Instruction accesses are not aliases.
Peripheral	Instruction fetches and data accesses are performed over the system bus.
Peripheral bit-band	Alias region. Data accesses are aliases. Instruction accesses are not aliases.
External RAM	Instruction fetches and data accesses are performed over the system bus.
External Device	Instruction fetches and data accesses are performed over the system bus.
Private Peripheral Bus	External and internal *Private Peripheral Bus* (PPB) interfaces. This memory region is *Execute Never* (XN), and so instruction fetches are prohibited. An MPU, if present, cannot change this.
System	System segment for vendor system peripherals. This memory region is XN, and so instruction fetches are prohibited. An MPU, if present, cannot change this.

5.9. Memory model

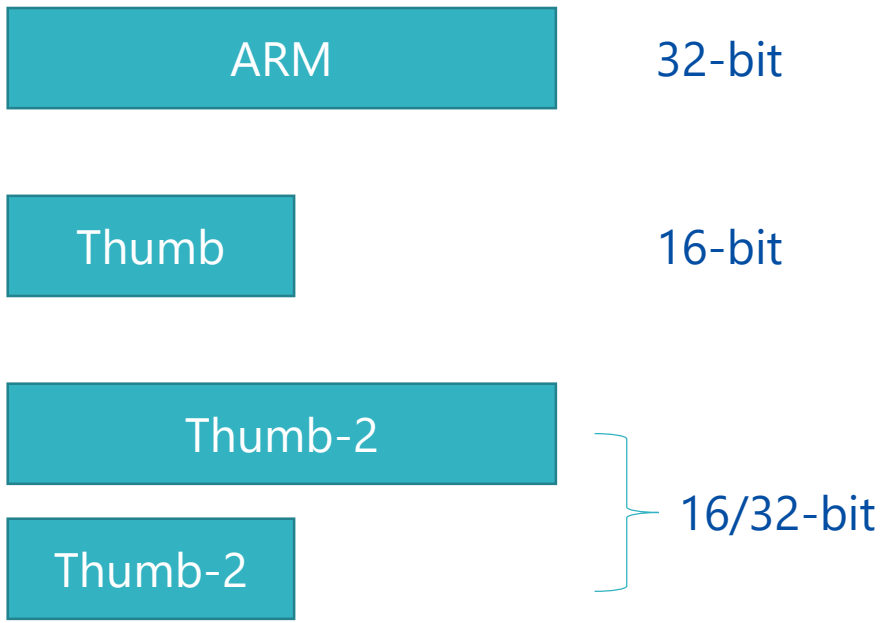
Memory map – Regions:



- **ICode memory interface :** Instruction fetches from Code memory space, 0x00000000 to 0x1FFFFFFC, are performed over the 32-bit AHB-Lite bus.
- **DCode memory interface** : Data and debug accesses to Code memory space, 0x00000000 to 0x1FFFFFFF, are performed over the 32-bit AHB-Lite bus.
- **System interface :** Instruction fetches and data and debug accesses to address ranges 0x20000000 to 0xDFFFFFFF and 0xE0100000 to 0xFFFFFFFF are performed over the 32-bit AHB-Lite bus.
- **Private Peripheral Bus (PPB) :** Data and debug accesses to external PPB space, 0xE0040000 to 0xE00FFFFF, are performed over the 32-bit Advanced Peripheral Bus (APB) bus. The *Trace Port Interface Unit* (TPIU) and vendor specific peripherals are on this bus.

5.10. ARM register set

- All instructions are 32 bits long
- Most instructions can be conditionally executed
- Load/Store instruction set – no direct manipulation of memory contents

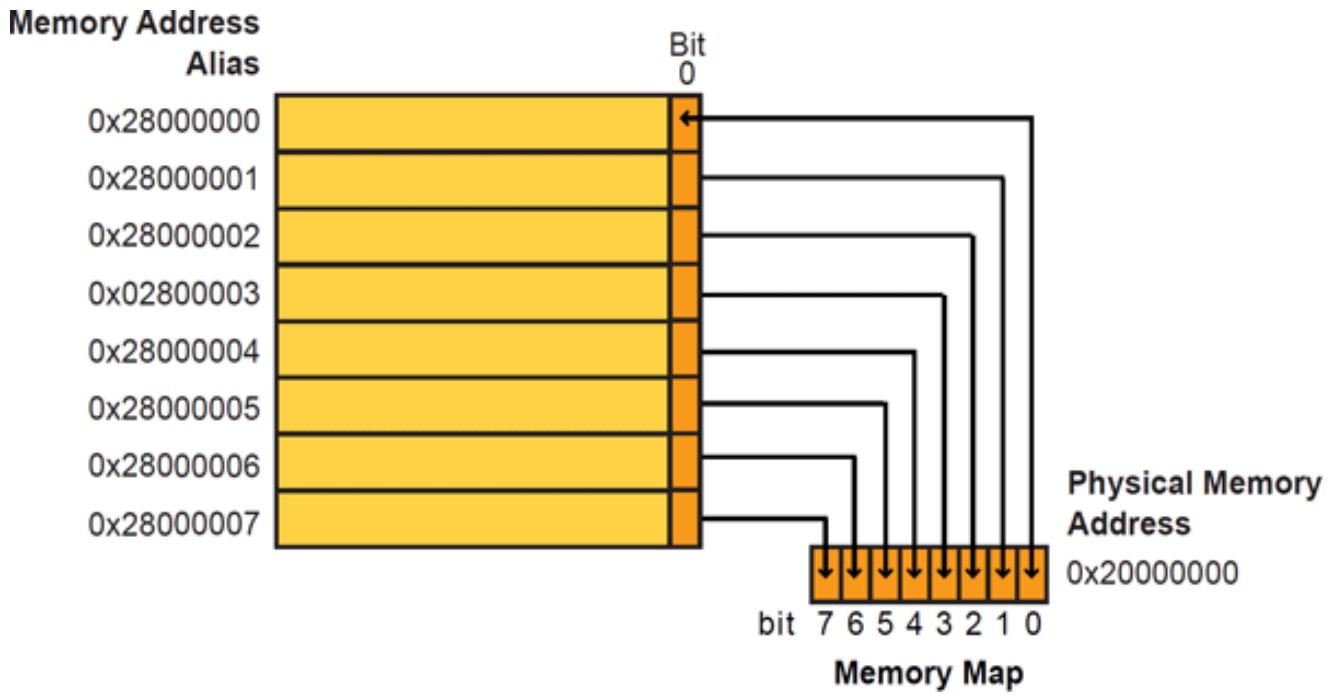


Operation
Pop
Semaphore
Branch
State change
Extend
Bit field
Reverse
Hint
Barriers

Operation
Move
Add
Subtract
Multiply
Divide
Saturate
Compare
Logical
Shift
Rotate
Count
Store
Push

5.11. Bit-banding

- Bit-banding is an optional feature of the Cortex-M4 processor.
- Bit-banding maps a complete word of memory onto a single bit in the bit-band region.
- For example, writing to one of the alias words sets or clears the corresponding bit in the bit-band region.

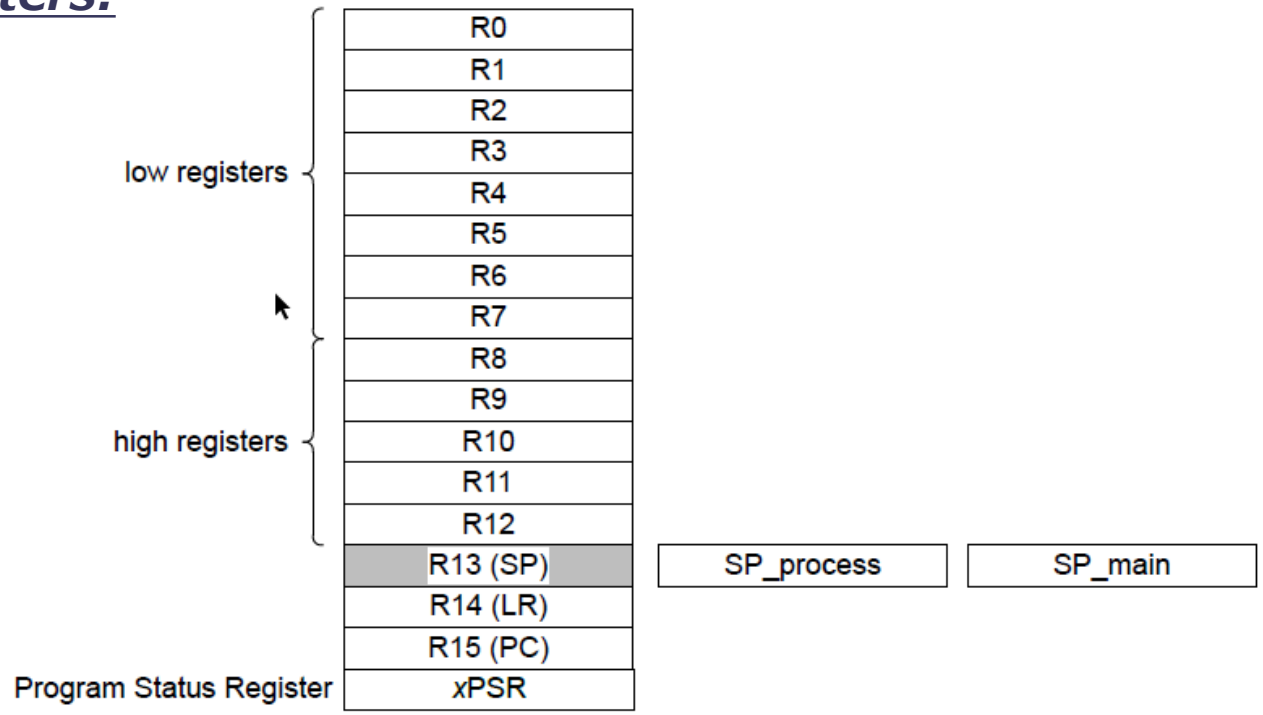


5.12. Processor core register summary

The processor has the following 32-bit registers:

- 13 general-purpose registers
  - Register R0-R7(low registers)
  - Register R8-R12(high registers)

- 3 registers with special usage
  - *Stack Pointer* (SP) - R13 alias of banked registers, SP_process and SP_main.
  - *Link Register* (LR) - R14.
  - *Program Counter* (PC) - R15.
- Special-purpose registers
  - *x*PSR is the *Program Status Registers*

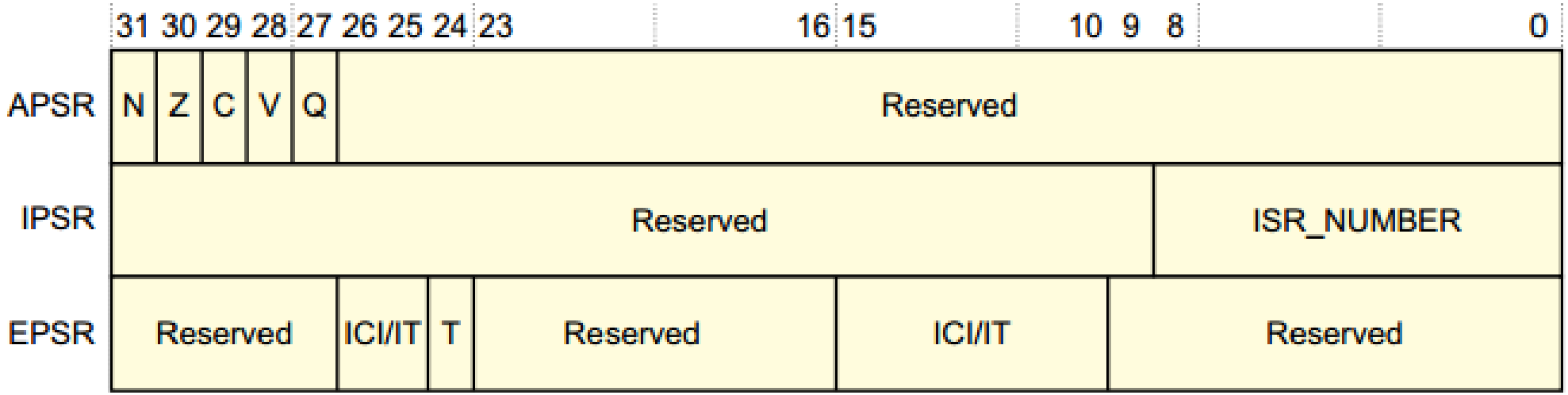


- ❖ SP: Handler mode always uses SP_main, but you can configure Thread mode to use either SP_main or SP_process.
- ❖ LR: The LR receives the return address from PC
- ❖ PC: It contains the current program address

5.12. Processor core register summary

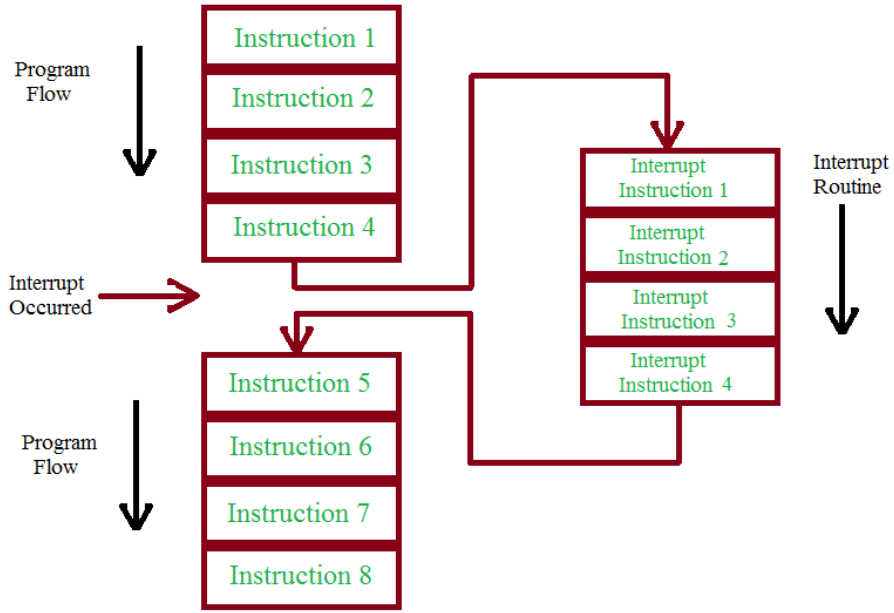
Program Status Register:

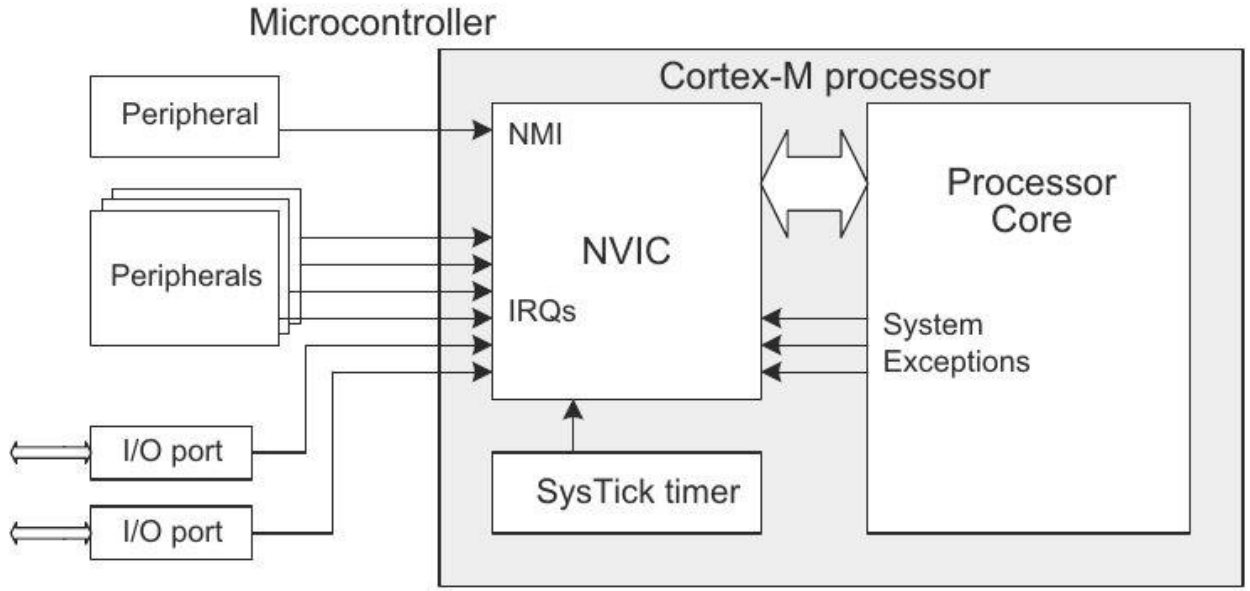
- Application Program Status Register (APSR)
- Interrupt Program Status Register (IPSR)
- Execution Program Status Register (EPSR)



5.13. Exceptions

- Exceptions are handled and prioritized by the processor and the NVIC.
- All exceptions are handled in Handler mode.





Technical documents

1. Cortex-M4 Technical Reference Manual - TRM
2. Cortex-M4 Integration and Implementation Manual – available as part of the Bill of Materials
3. Armv7-M Architecture Reference Manual - ARM
4. CoreSight ETM-M4 Technical Reference Manual – ETM
5. Cortex-M4-Processor-Datasheet


Thank you