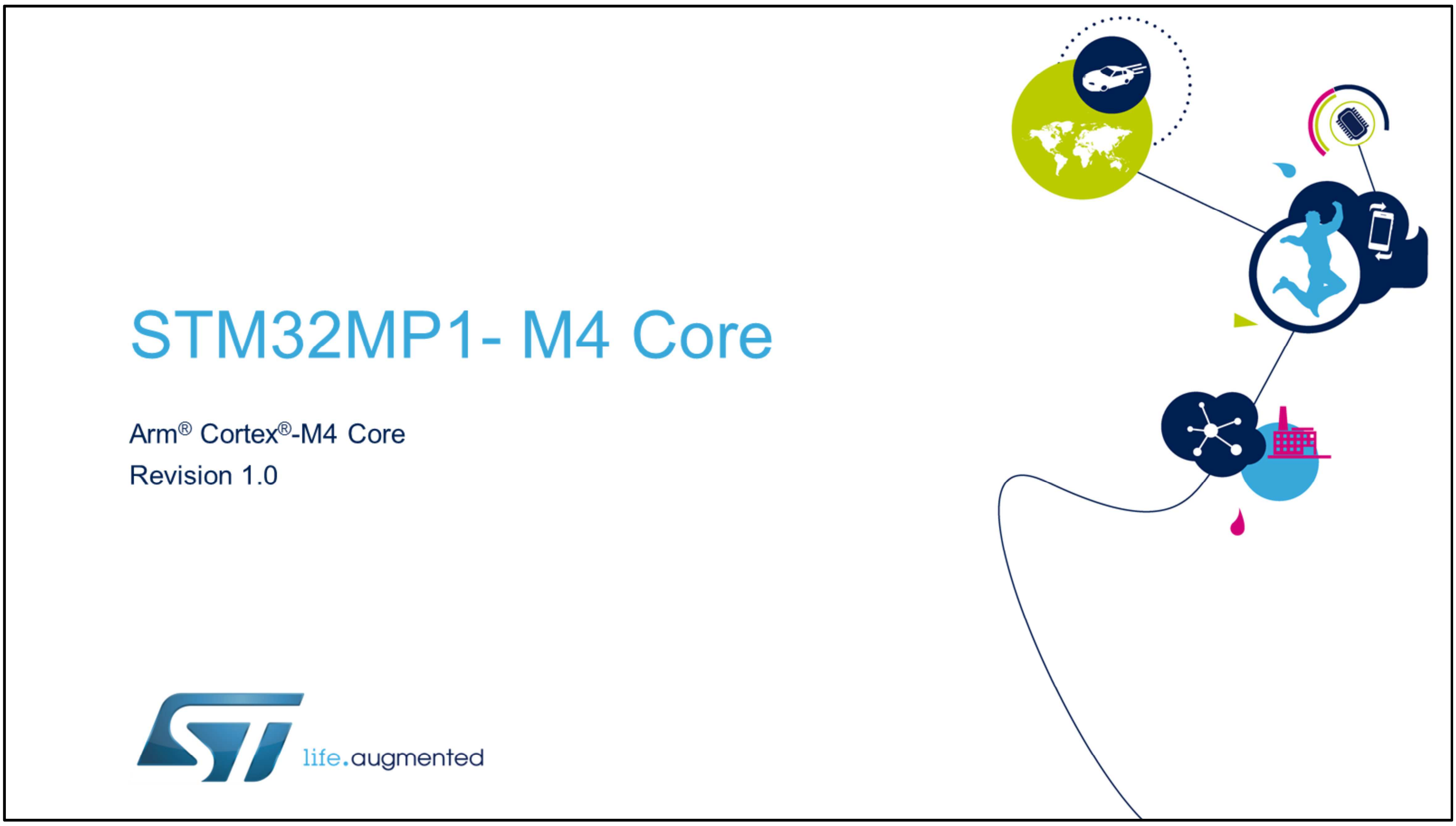


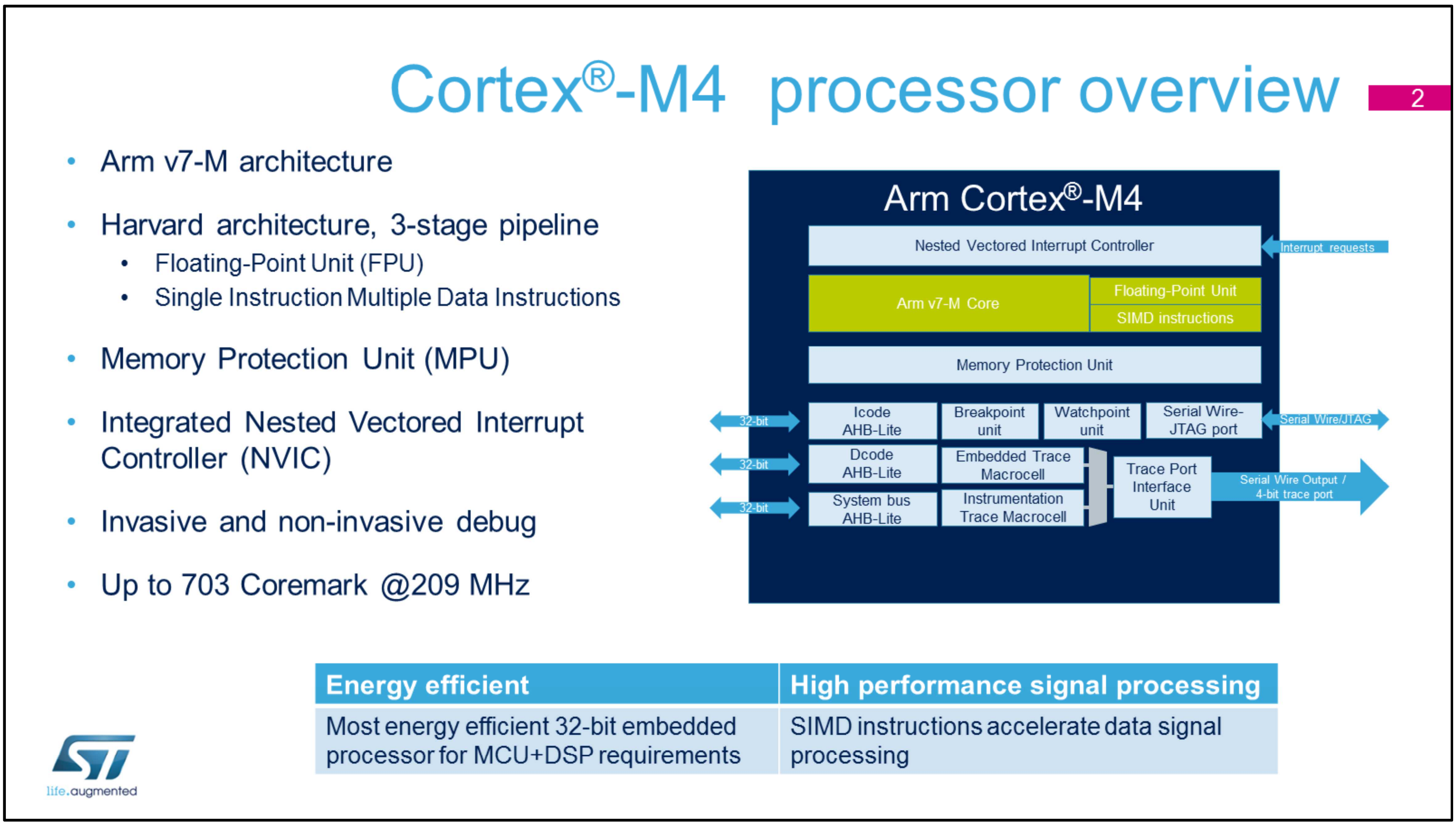
STM32MP1- M4 Core

Arm® Cortex®-M4 Core

Revision 1.0

life.augmented

Hello, and welcome to this presentation of the Arm® Cortex®-M4 core which is embedded in the STM32MP1 microprocessor series.



Cortex®-M4 processor overview

2

- Arm v7-M architecture
- Harvard architecture, 3-stage pipeline
  - Floating-Point Unit (FPU)
  - Single Instruction Multiple Data Instructions
- Memory Protection Unit (MPU)
- Integrated Nested Vectored Interrupt Controller (NVIC)
- Invasive and non-invasive debug
- Up to 703 Coremark @209 MHz

Arm Cortex®-M4

Nested Vectored Interrupt Controller

Interrupt requests

Arm v7-M Core

Floating-Point Unit

SIMD instructions

Memory Protection Unit

32-bit

32-bit

32-bit

Icode AHB-Lite

Dcode AHB-Lite

System bus AHB-Lite

Breakpoint unit

Watchpoint unit

Serial Wire-JTAG port

Serial Wire/JTAG

Embedded Trace Macrocell

Instrumentation Trace Macrocell

Trace Port Interface Unit

Serial Wire Output / 4-bit trace port

Energy efficient	High performance signal processing
Most energy efficient 32-bit embedded processor for MCU+DSP requirements	SIMD instructions accelerate data signal processing

life.augmented

The Cortex®-M4 core is part of the Arm Cortex®-M group of 32-bit RISC cores. It implements the Arm v7-M architecture and features a 3-stage pipeline. In addition to scalar integer instructions, it also supports a single precision floating-point unit and SIMD integer instructions, useful to improve the performance of DSP algorithms.

It offers up to 703 Coremarks when running at 209 MHz.

The Cortex®-M4 has three AHB-Lite master ports, enabling concurrent instruction and data transactions.

Interrupts received from STM32MP1 peripherals are handled by the Nested Vectored Interrupt Controller (NVIC).

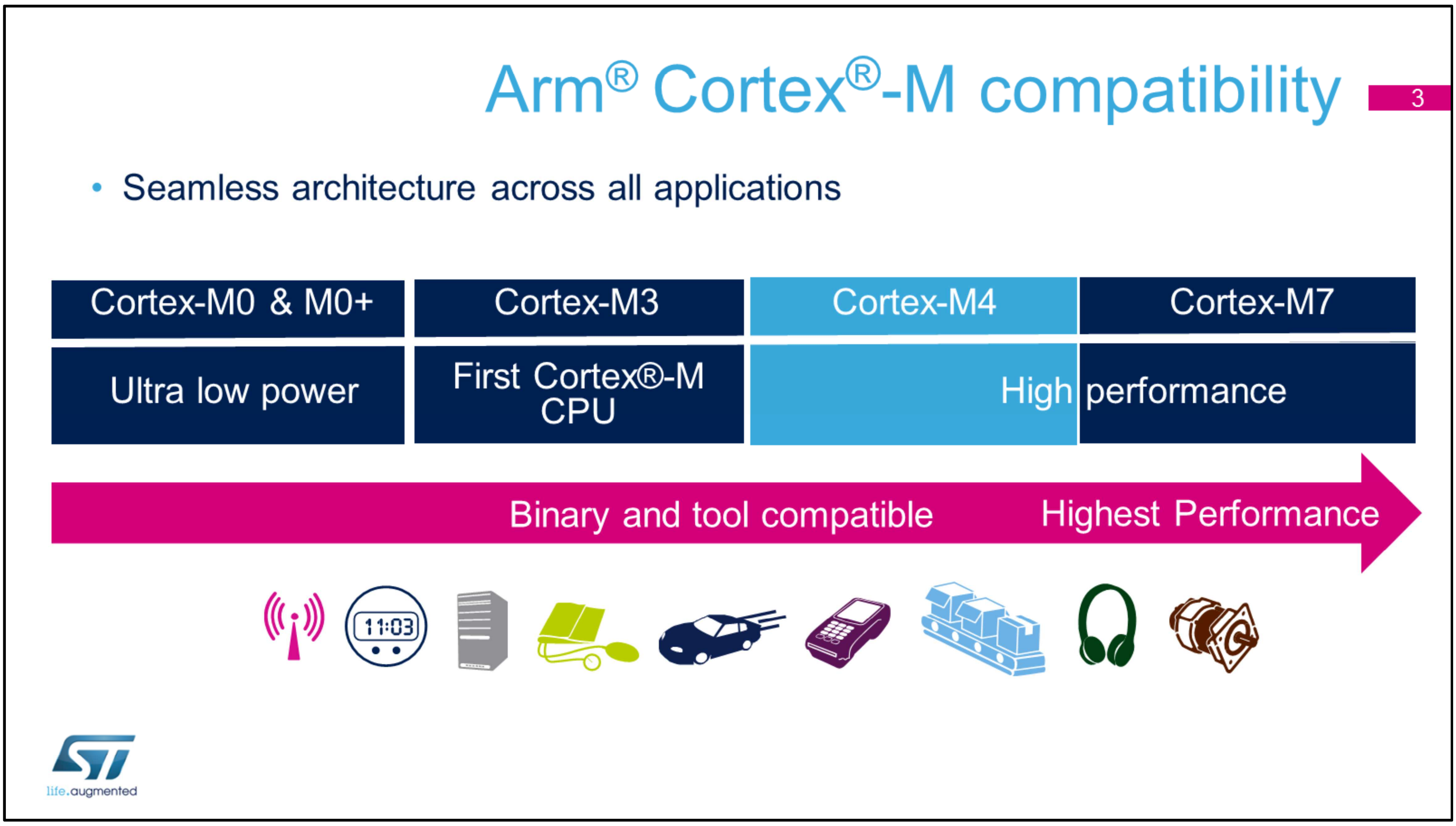
The Memory Protection Unit (MPU) is in charge of assigning attributes and access permissions to instruction and data requests initiated by the core.

Many debug units are implemented.

Two protocols can be used to communicate between the Serial Wire or JTAG debug port(SWJ-DP) and the external debug probe: either serial wire or JTAG.

Invasive debug is performed by means of the breakpoint and watchpoint units.

Regarding non-invasive debug, the Cortex®-M4 supports two real time trace capabilities: the Embedded Trace Macrocell (ETM) and the Instrumentation Trace Macrocell (ITM). Trace packets are output to the external Trace Port Analyzer through the Trace Port Interface Unit (TPIU).



Arm® Cortex®-M compatibility

- Seamless architecture across all applications

Cortex-M0 & M0+	Cortex-M3	Cortex-M4	Cortex-M7
Ultra low power	First Cortex®-M CPU	High performance	

Binary and tool compatible Highest Performance

life.augmented

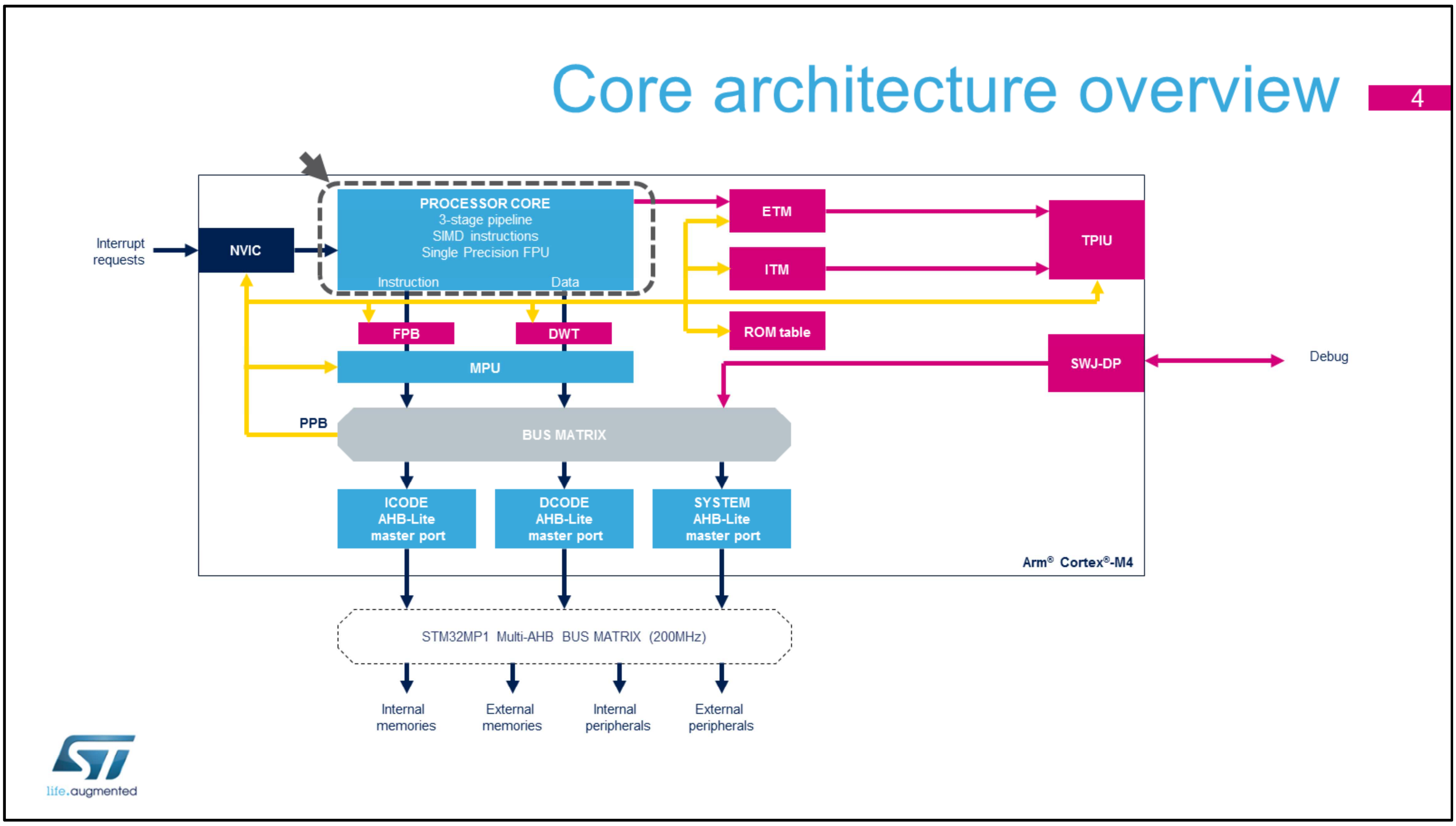
STM32MP1 microprocessors integrate an Arm® Cortex®-M4 core in order to benefit from the powerful performance of its 32-bit processor architecture and particularly high level of deterministic processing.

All Cortex®-M CPUs have a 32-bit architecture.

The Cortex®-M3 was the first Cortex®-M CPU released by Arm.

Then Arm decided to distinguish two product lines: high performance and low power, while maintaining the compatibility between them.

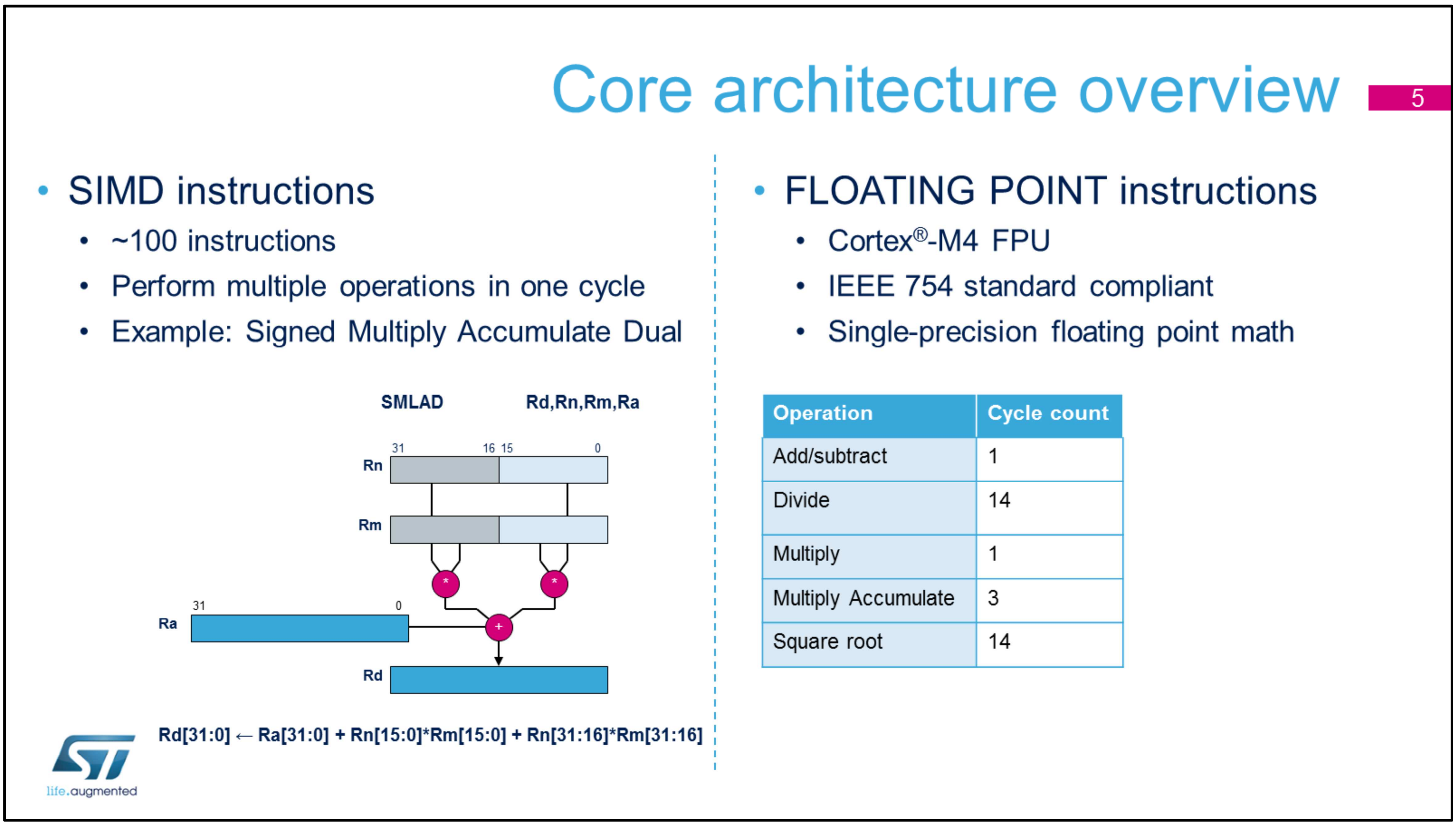
The Cortex®-M4 belongs to the high performance product line.



Core architecture overview

The processor core implements a Harvard architecture, as it supports concurrent instruction fetch and data load/store transactions.

The instruction pipeline features three stages: fetch, decode and execute. Conditional branch execution is accelerated by early fetching the target instruction.



Core architecture overview

- SIMD instructions
  - ~100 instructions
  - Perform multiple operations in one cycle
  - Example: Signed Multiply Accumulate Dual

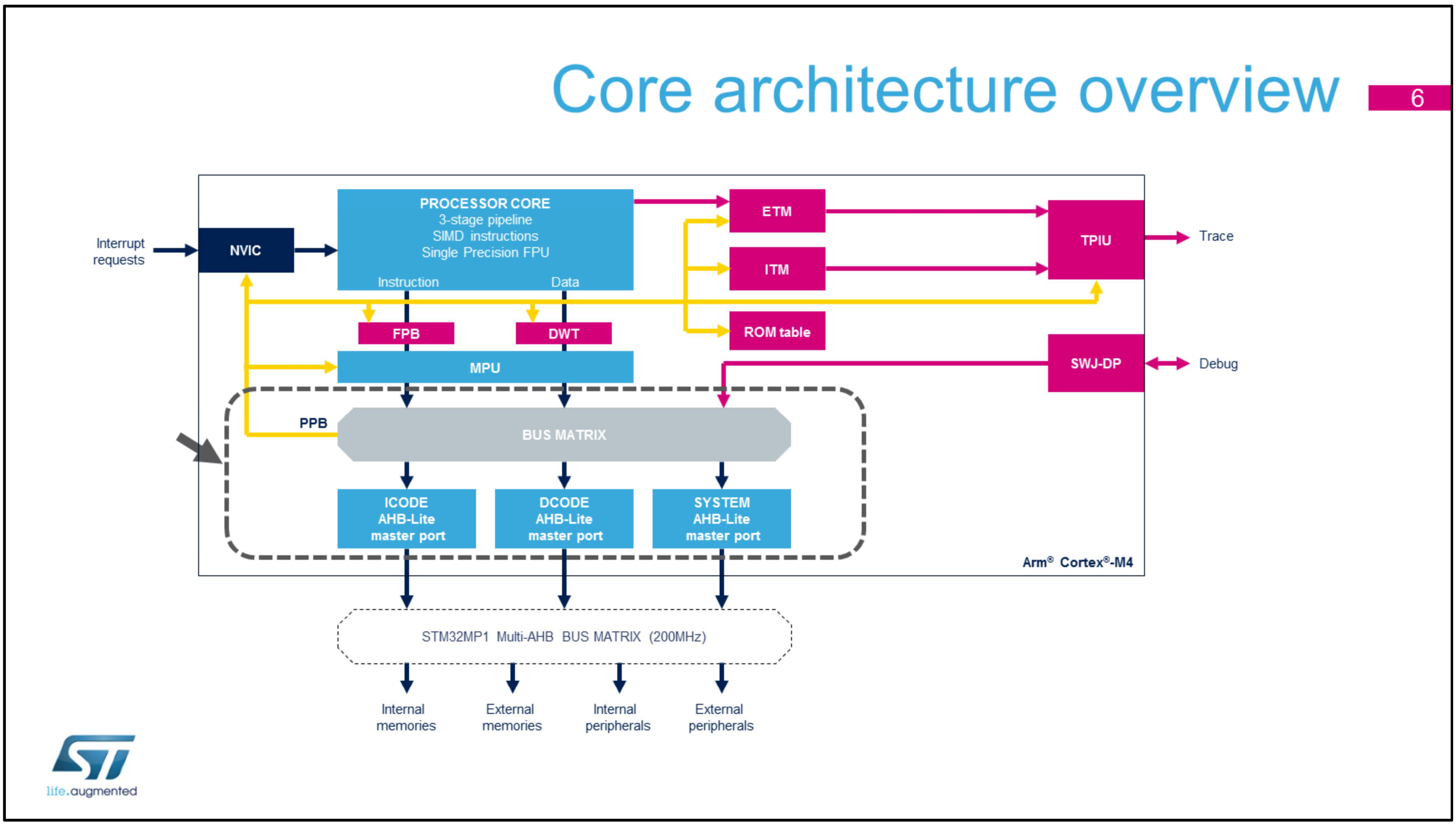
Rd[31:0] ← Ra[31:0] + Rn[15:0]*Rm[15:0] + Rn[31:16]*Rm[31:16]

- FLOATING POINT instructions
  - Cortex®-M4 FPU
  - IEEE 754 standard compliant
  - Single-precision floating point math

Operation	Cycle count
Add/subtract	1
Divide	14
Multiply	1
Multiply Accumulate	3
Square root	14

life.augmented

SIMD techniques operate with packed data. For instance, two 12-bit samples acquired with the ADC can be stored in the two halfwords of the same 32-bit register. In the example described in this slide, two pairs of samples are multiplied and then accumulated into a destination register. Since data signal processing is based on sum of products, SIMD instructions contribute to increase the performance with regard to regular scalar fixed-point instructions.
The Cortex®-M4 core embedded in the STM32MP1 microprocessor implements the optional single precision floating point unit, which is compatible with the IEEE754 standard. Add, Subtract and Multiply instructions take one clock to execute, Multiply Accumulate
instruction takes 3 clocks, Divide and Square Root instructions take 14 clocks.



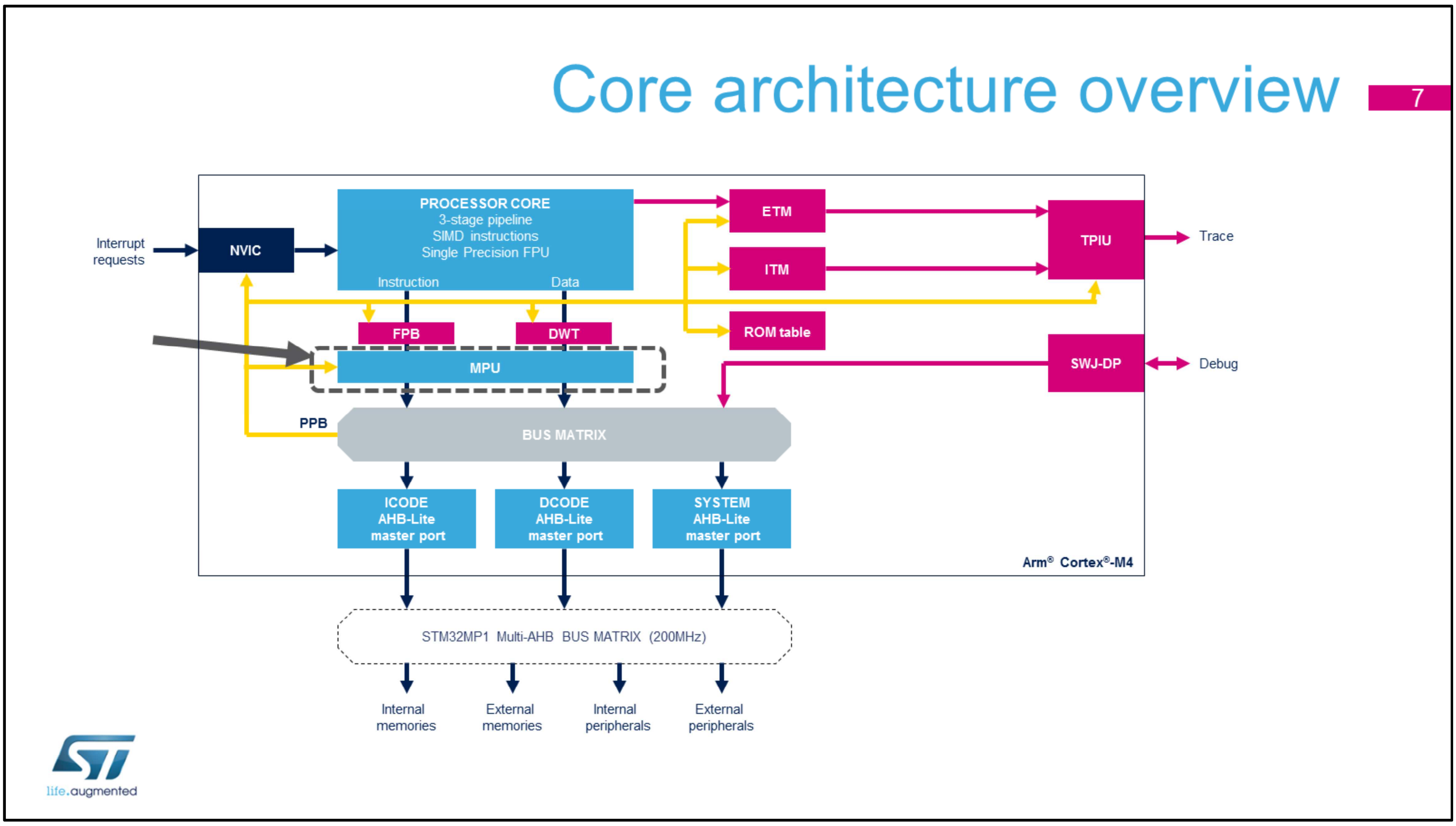
Core architecture overview

6

The Cortex®-M4 has neither a cache nor an internal RAM. Consequently any instruction fetch transaction and data access is steered to the internal bus matrix. This bus matrix selects the output AHB-lite master port according to the address and the access type, instruction or data. Three AHB transactions can be in progress at a time, for instance:

- An instruction access from flash memory using the ICODE master port
- A constant data access from flash memory using the DCODE master port
- An SRAM access using the SYSTEM master port.

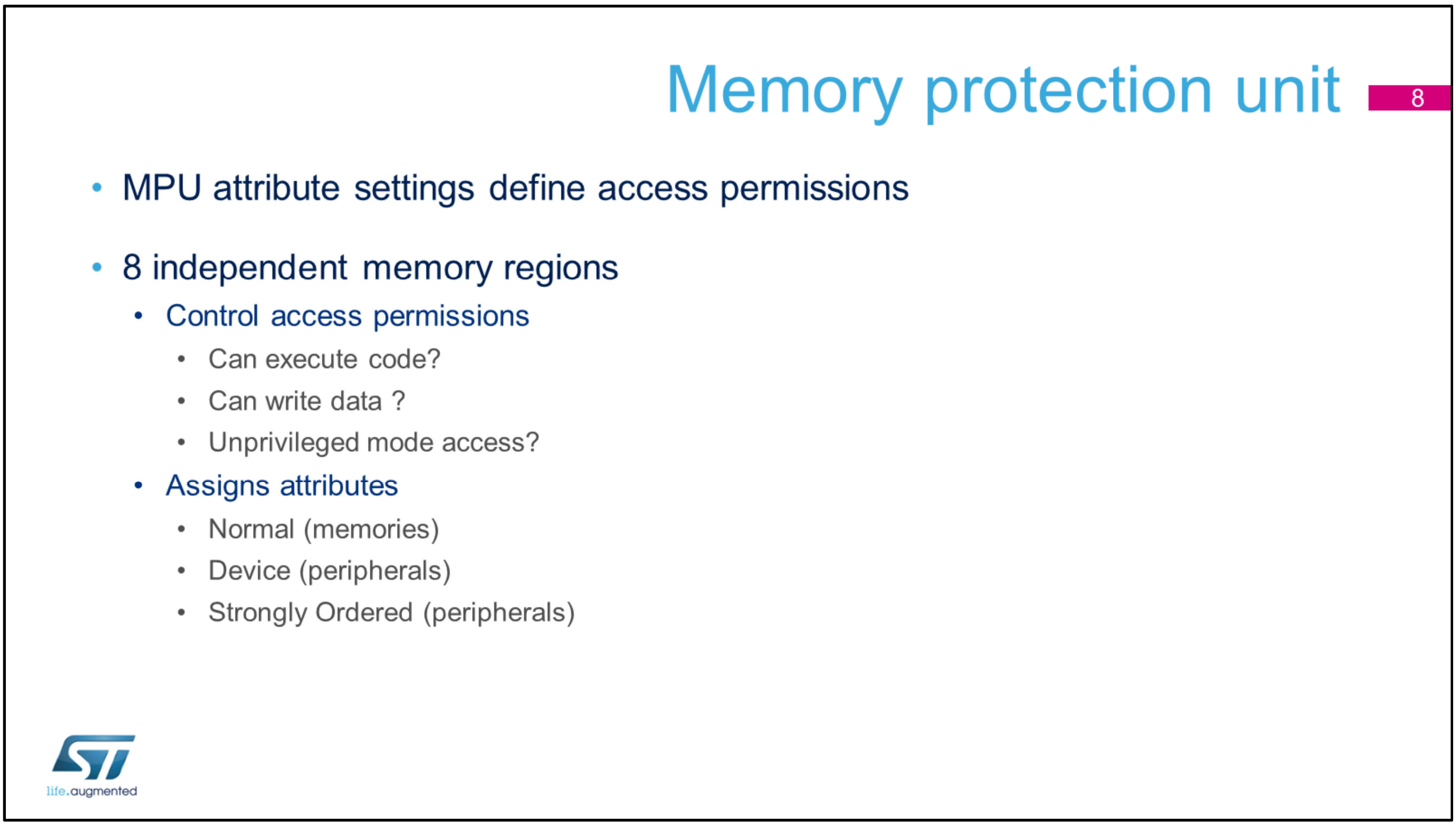
The Cortex®-M4's bus matrix is connected to the STM32MP1 Multi-AHB bus matrix, enabling the CPU to access memories and peripherals. Since transactions are pipelined on AHB-Lite, the best throughput is 32 bits of data or instructions per clock, with a minimum 2-clock latency. One of the output of the Cortex®-M4's bus matrix is the Private Peripheral Bus (PPB), which is internal to the CPU. It is used to access memory-mapped registers present in NVIC, MPU and debug units.



Core architecture overview

7

In the Cortex®-M4 core, the Memory Protection Unit (MPU) is used to protect address ranges according to the configured access permissions. When enabled, it intercepts any access initiated by the processor core.



Memory protection unit

- MPU attribute settings define access permissions
- 8 independent memory regions
  - Control access permissions
    - Can execute code?
    - Can write data ?
    - Unprivileged mode access?
  - Assigns attributes
    - Normal (memories)
    - Device (peripherals)
    - Strongly Ordered (peripherals)

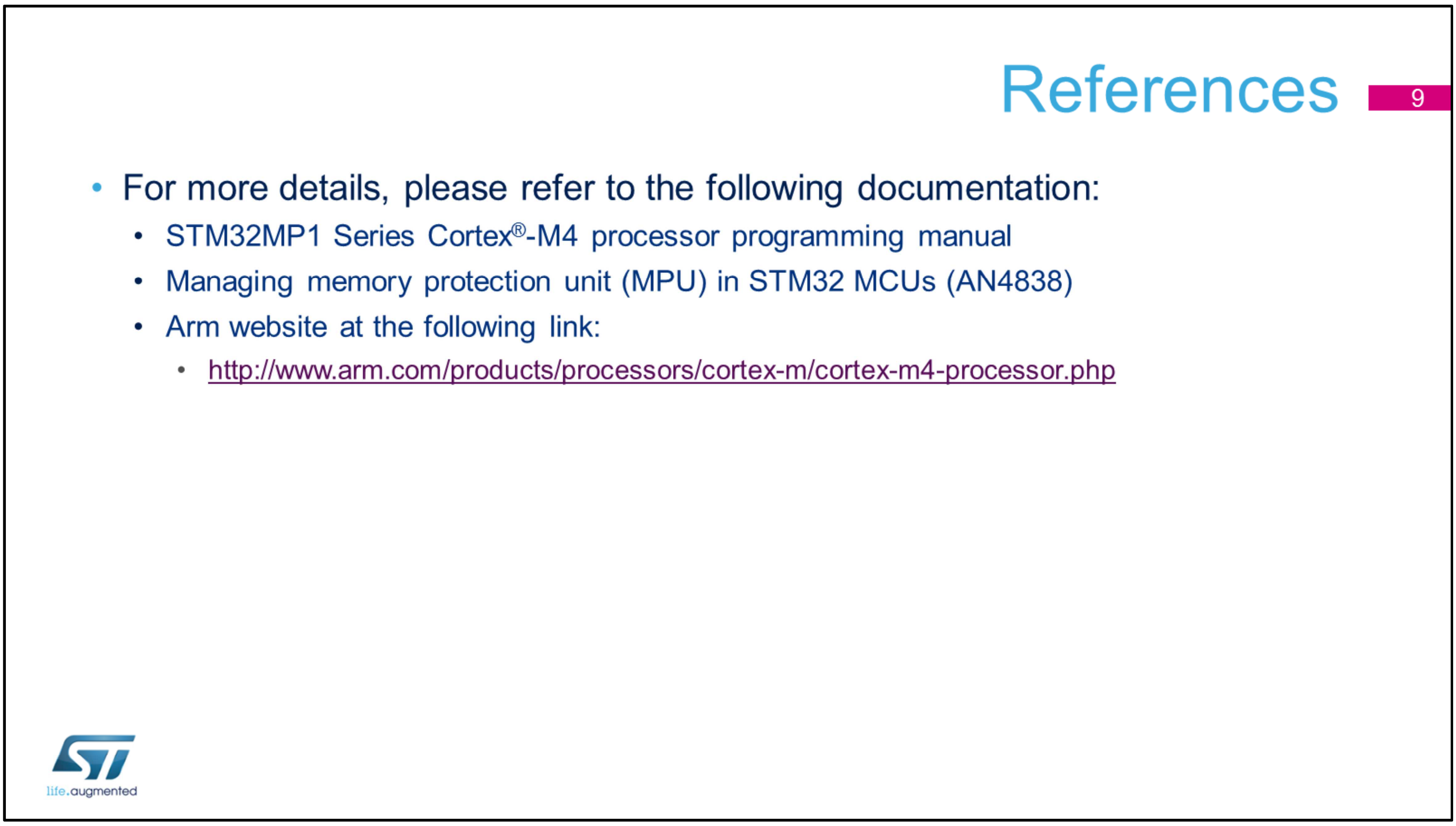
The Memory protection unit (MPU) in the STM32MP1 microprocessor offers support for eight independent memory regions, with independent configurable access permissions for:

- Access permission: allowed or not read/write in privileged/unprivileged mode,
- Execution permission: executable region or region prohibited for instruction fetch.

The MPU is also in charge of assigning attributes to regions, called Normal, Device and Strongly Ordered.

Normal is used to map memories.

Device and Strongly Ordered attributes are used to map peripherals. The difference between them is the ability to buffer data during the peripheral access. The Device attribute enables write posting while a store to a Strongly Ordered region stalls the pipeline until the response is received from the targeted peripheral.



References

- For more details, please refer to the following documentation:
  - STM32MP1 Series Cortex®-M4 processor programming manual
  - Managing memory protection unit (MPU) in STM32 MCUs (AN4838)
  - Arm website at the following link:
    - http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php

The NVIC and debug units are described in separate presentations.
For more details, please refer to these application notes and the Cortex®-M4 programming manual available on www.st.com website.
Also visit the Arm website where you can find more information about the Cortex®-M4 core.