



# STM32MP15 RAM mapping



# STM32MP15 RAM mapping

Stable: 07.04.2020 - 12:20 / Revision: 25.03.2020 - 10:06

## Contents

1 Overview .....	2
2 Arm core characteristics .....	2
<b>2.1 Reset address .....</b>	<b>2</b>
<b>2.2 Cortex-M4 multiple ports .....</b>	<b>2</b>
3 Memory mapping .....	3
<b>3.1 Overall memory mapping .....</b>	<b>3</b>
<b>3.2 Zoom in the Cortex-A7/Cortex-M4 shared memory .....</b>	<b>4</b>

## 1 Overview

This article shows the default memory mapping defined by STMicroelectronics in STM32MPU Embedded Software. It uses a subset of all memory regions that are exposed at hardware level: customers may use other memory regions or aliases that are not shown here but are described in the [STM32MP15 reference manuals](#).

## 2 Arm core characteristics

Integration of Arm cores sets some constraints on the device memory mapping: the main ones are listed in this article.

### 2.1 Reset address

Arm<sup>®</sup> Cortex<sup>®</sup> cores start running from address 0x00000000 on reset, which is why this address respectively points to:

- The ROM code on the Cortex-A7 side. This read-only memory embeds the boot code that is executed when the platform boots (and executes the [boot chain](#)) or wakes up from [low power STANDBY mode](#).
- The Retention RAM on the Cortex-M4 side. This needs to be loaded by the Cortex-A7 before releasing the Cortex-M4 reset (in the [RCC](#)) and getting it running. This is done by [Linux coprocessor management](#), by default.

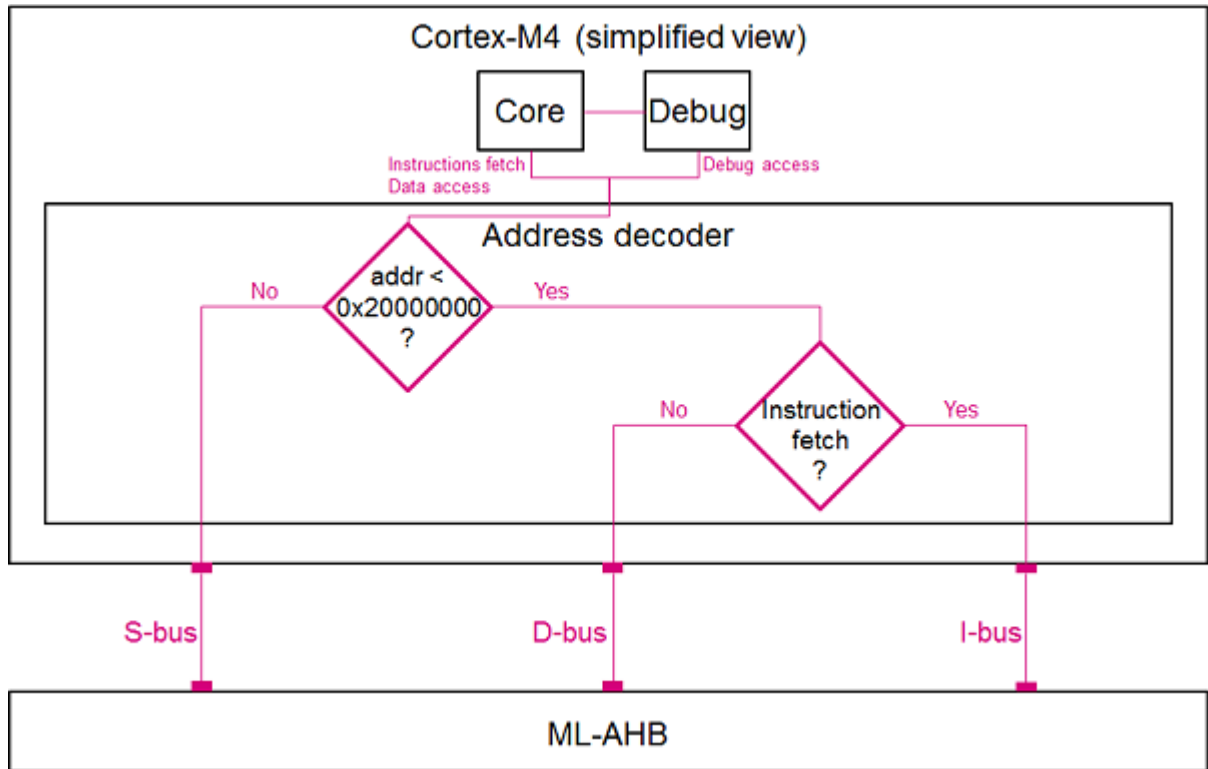
Note: since the Cortex-A7 has its ROM code mapped at address 0x00000000, it uses a hardware alias to access the retention RAM at address 0x38000000

### 2.2 Cortex-M4 multiple ports

The Cortex-M4 is connected to the interconnect (ML-AHB) via three ports, listed below and shown in the following figure:

- I-bus is used to fetch code instructions in the 0x00000000--0x1FFFFFFF address range
- D-bus is used to read/write data in the 0x00000000--0x1FFFFFFF address range

- S-bus is used for all accesses in the 0x20000000--0xFFFFFFFF address range; all STM32MP15 internal peripherals registers are mapped in this range.



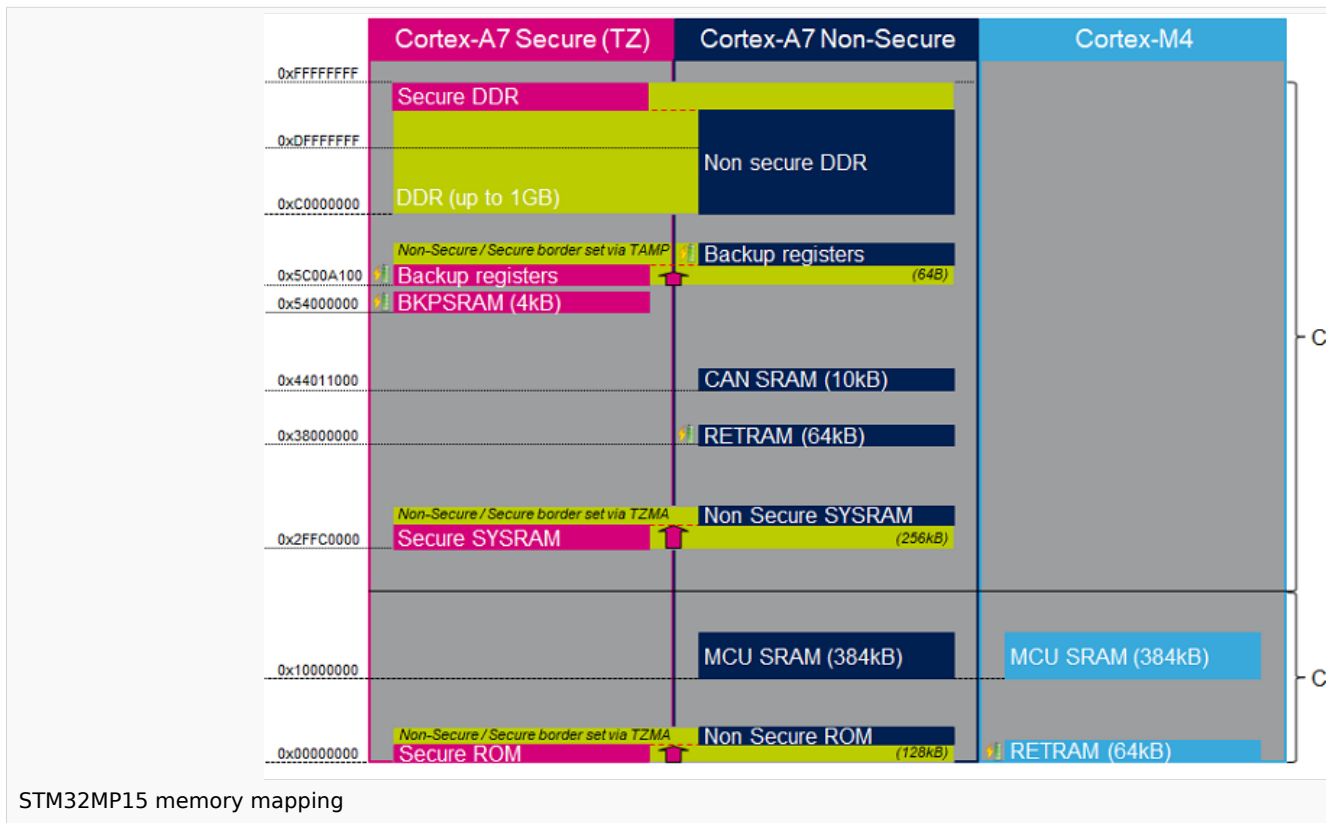
Balancing the Cortex-M4 firmware accesses among those ports allows tuning of the system performance, which is why the MCU SRAM is defined in the first address range (from 0x10000000), but is also visible in the second range (from 0x30000000) in the STM32MP15 reference manuals.

Nevertheless, it is important to notice that the Cortex-M4 embedded in the STM32MP15 only allows hardware breakpoints to be set on the address range covered by the I-bus. Thus any code accessed via the S-bus has to be debugged through software breakpoints.

## 3 Memory mapping

### 3.1 Overall memory mapping

The memory mapping below is a subset of all regions that are exposed at hardware level: it shows the default configuration used in OpenSTLinux but the customer may choose a different mapping to take advantage of other address ranges defined in STM32MP15 reference manuals.

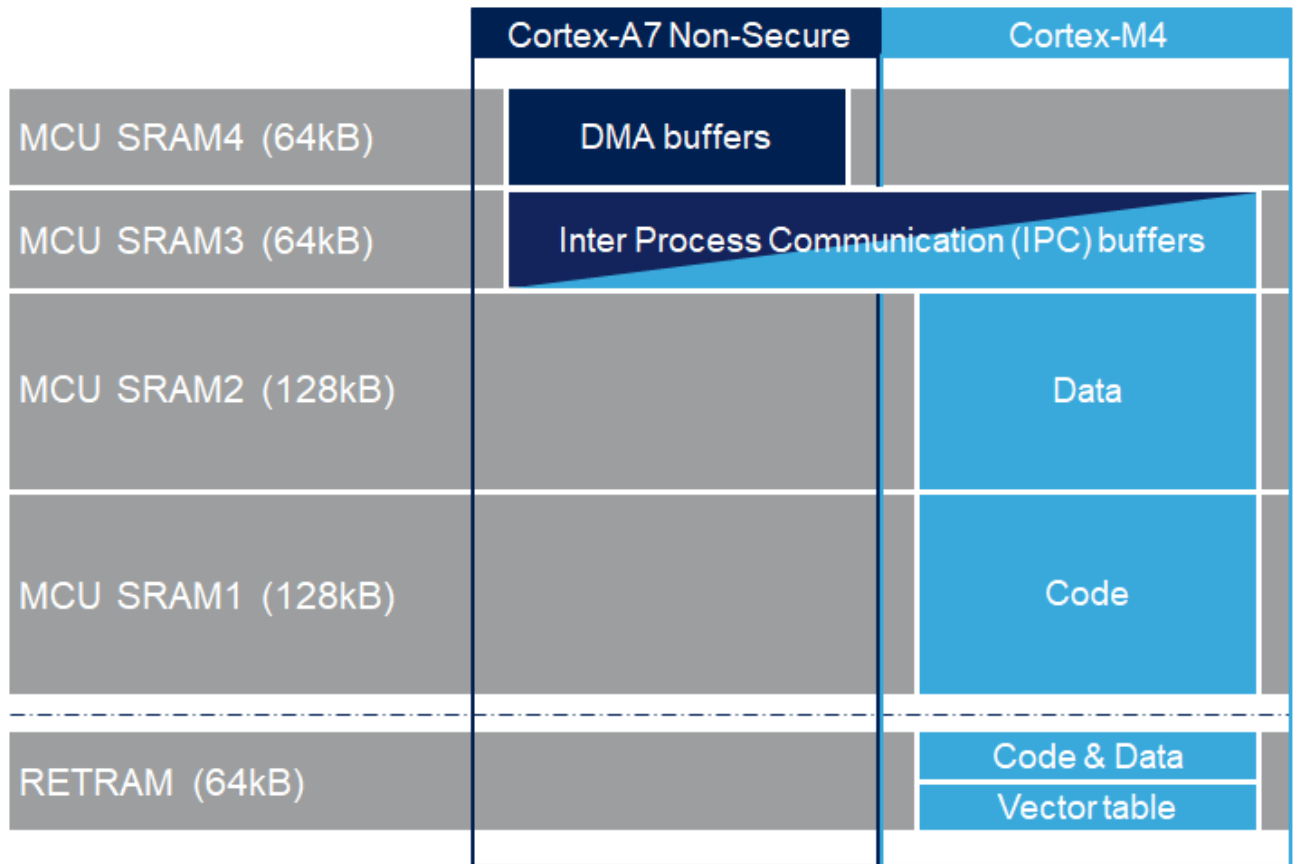


### 3.2 Zoom in the Cortex-A7/Cortex-M4 shared memory

The figure below is a zoom of the RAM areas that are shared between the Cortex-A7 non-secure and the Cortex-M4. This mapping is STMicroelectronics' default implementation that can be freely adapted by customers to fit to other needs.



STM32MP15 RAM mapping



Read Only Memory

Random Access Memory (Early computer memories generally had serial access. Memories where any given address can be accessed when desired were then called "random access" to distinguish them from the memories where contents can only be accessed in a fixed order. The term is used today for volatile random-access semiconductor memories.)

Advanced High-performance Bus