Task – 4 Research

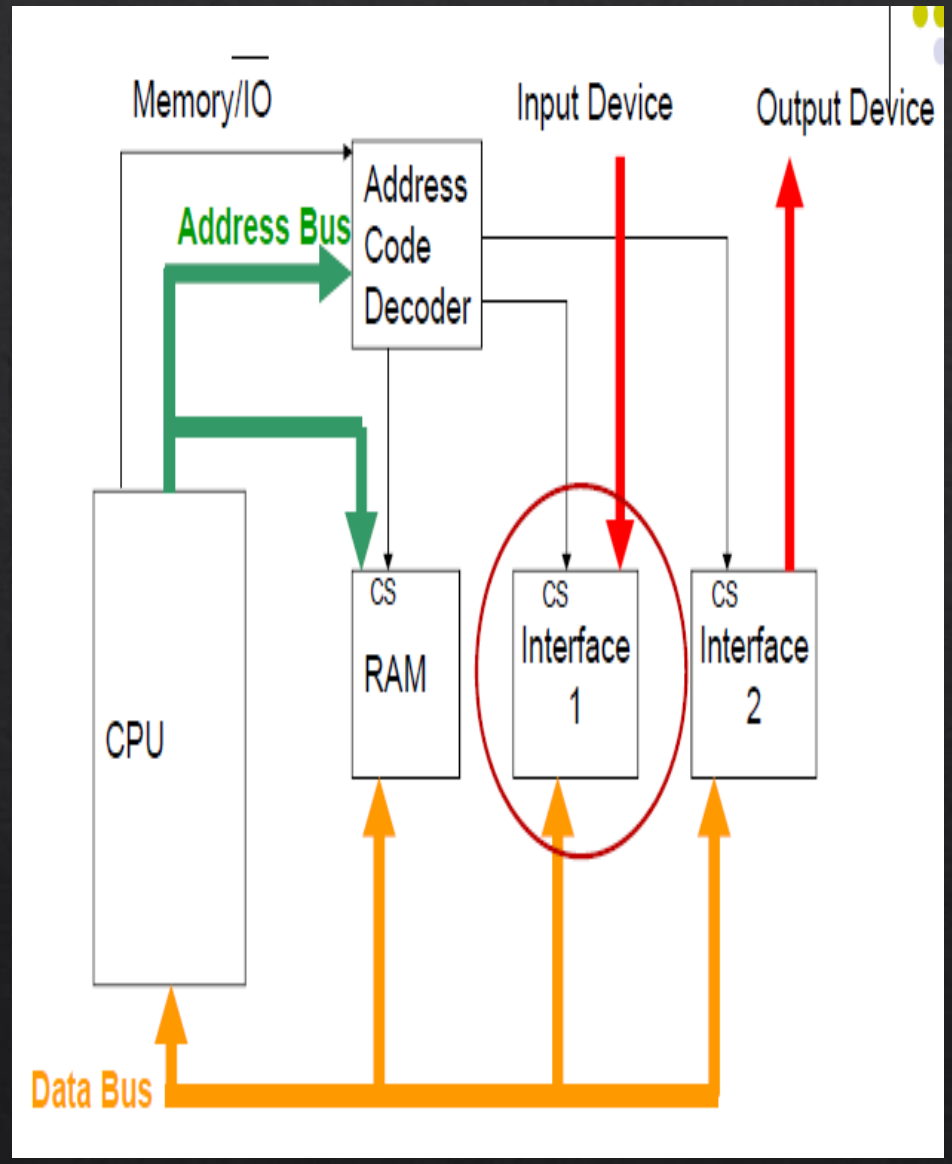
Memory mapping

Memory Mapping means manually assigning specific memory blocks to hardware components or memory regions like RAM, ROM, or Flash. This allows the CPU to access these devices directly, as if they were normal variables in memory.

It works like a map telling the CPU where each hardware device is, so it can read/write to their registers just like normal variables.

Types of memory mapping :

- Memory-Mapped I/O External devices (like LEDs, motors, etc.) are controlled by reading/writing to specific memory addresses. The CPU uses regular memory instructions to control hardware.
- Port-Mapped I/O devices are accessed through special IN and OUT instructions (used in old intel CPUs where it separates I/O space from memory space)



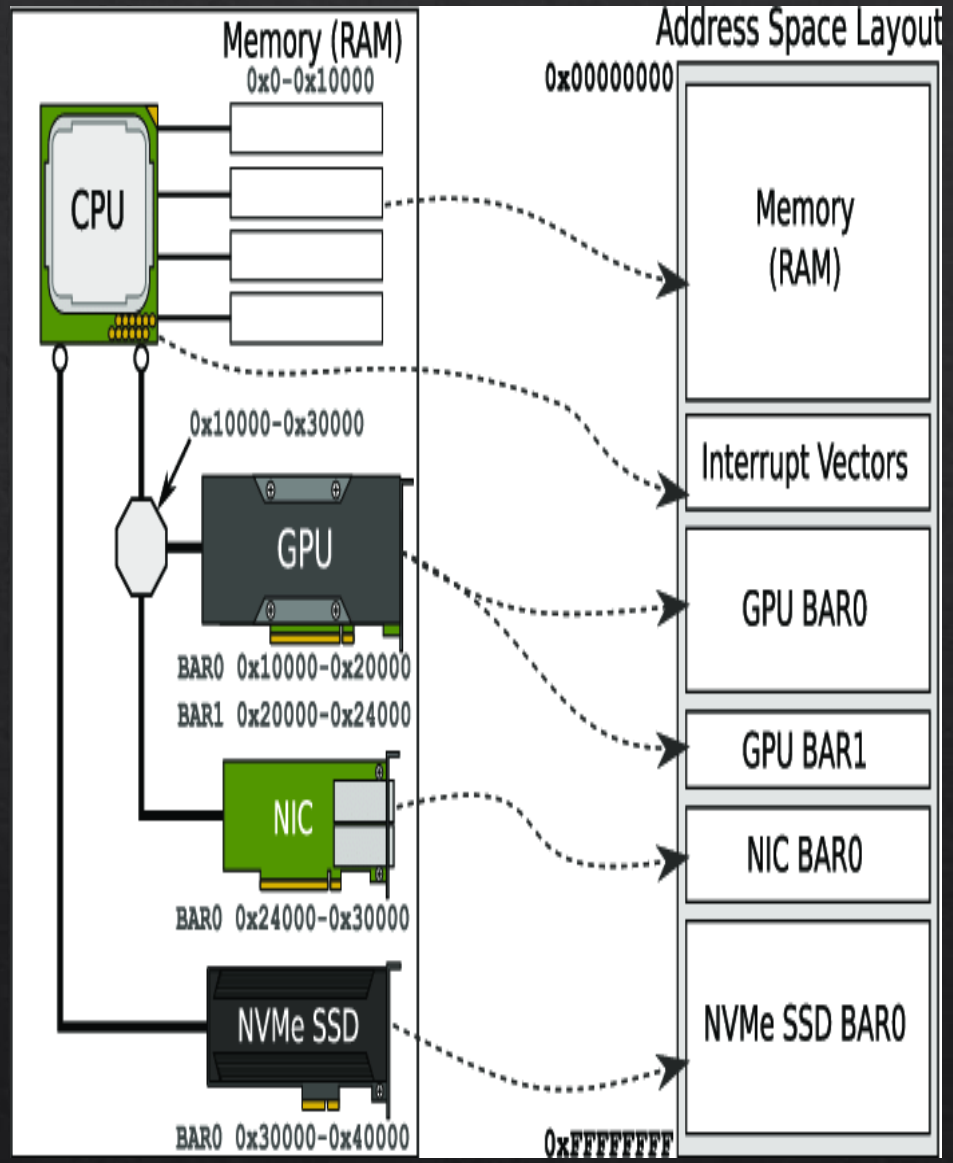
Memory mapping

Allocation of Addresses :

- **allocating a specific ranges of addresses for different purposes ex . A portion for RAM , a portion for ROM and another for controlling peripherals (same concept as stack, heap and static memory in software)**
- Allocating these ranges in microcontrollers is usually fixed determined by hardware design however some systems allow for dynamic allocating during run time to meet demands

CPU interaction with memory map :

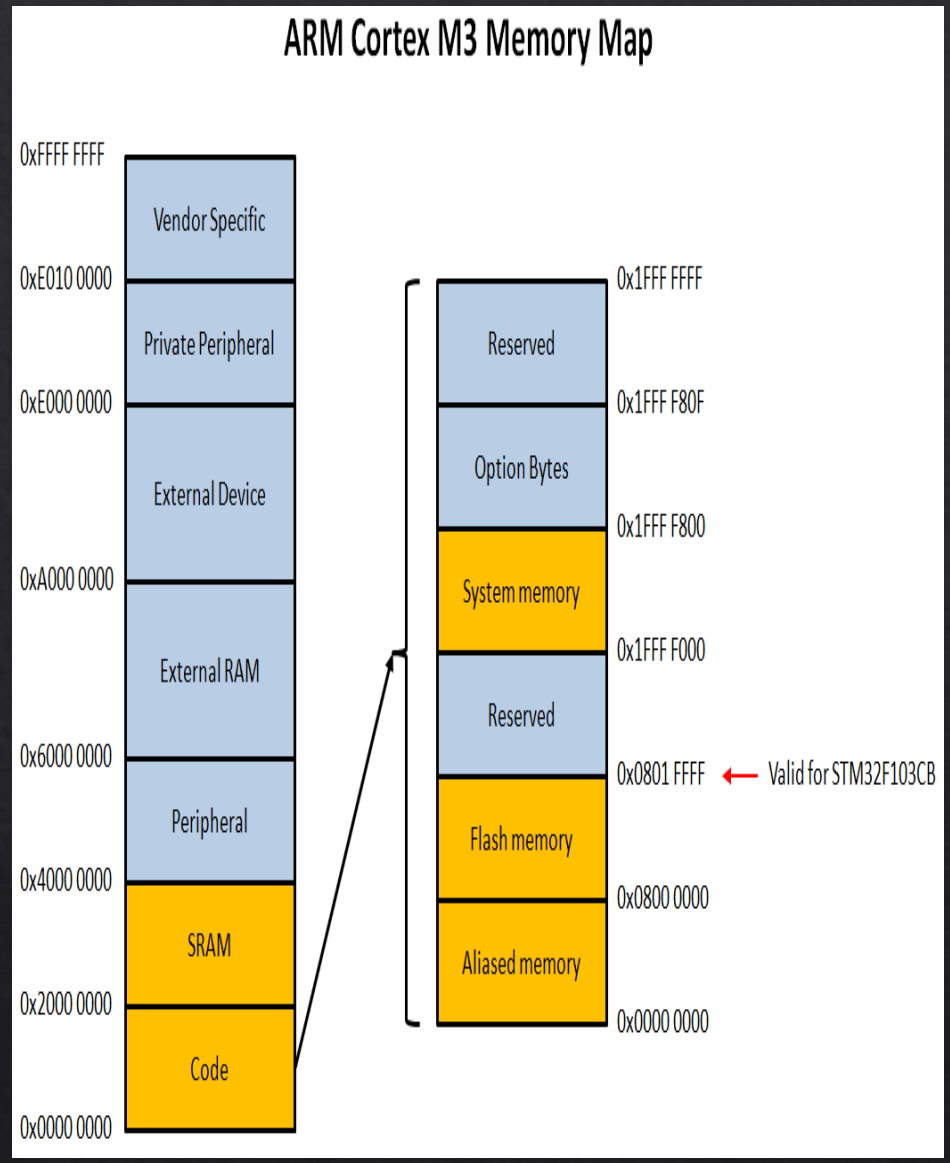
- The CPU sends a write/read signal and according to the address it's sent to the already-defined map routes this signal to the correct location



Memory regions in STM32F103C8

Code region : contains executable code and is divided into

- Main flash memory :
  - stores user code
  - Non-Volatile : the code remains after resetting or powering off
  - Most of it is used for firmware and bootloader
- System memory
  - Contains bootloader –provided by STM32-
  - Used for firmware upgrades



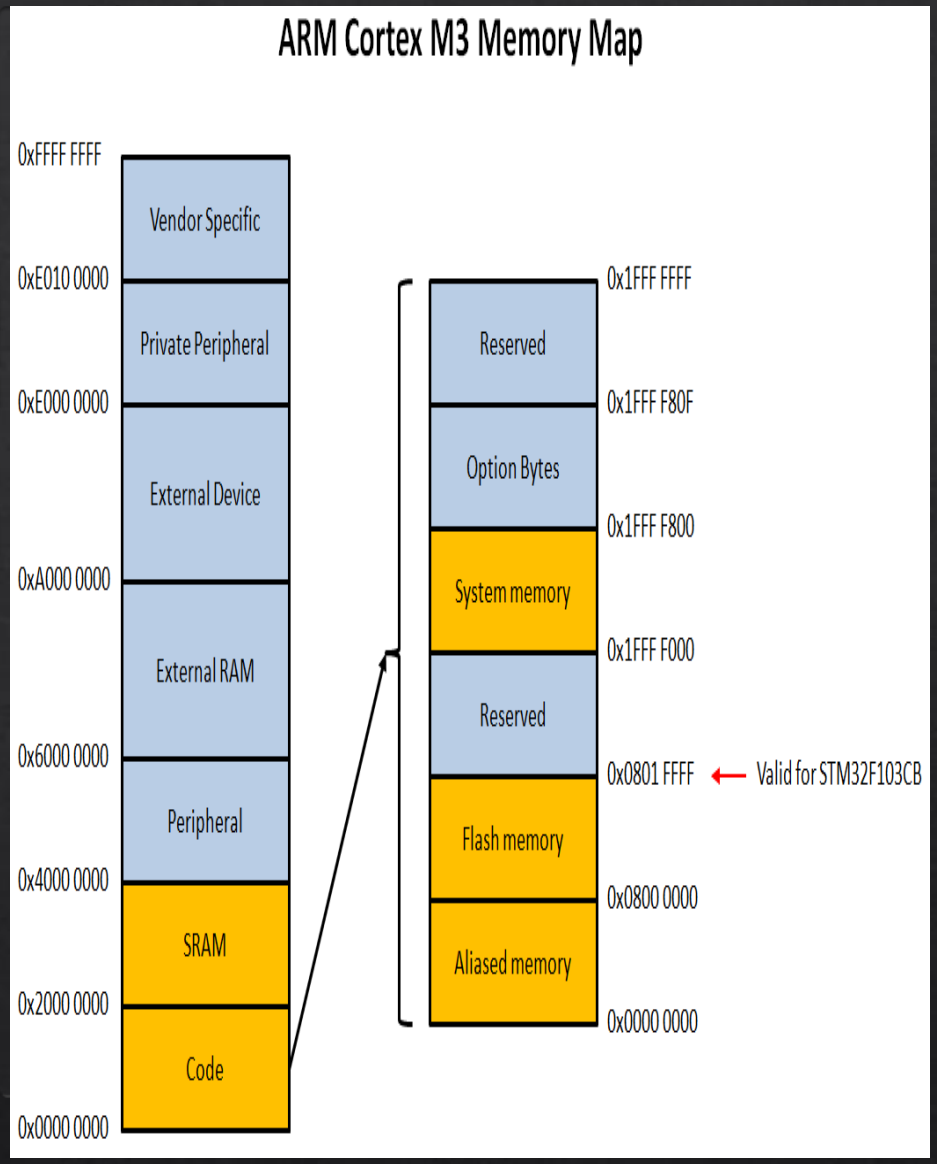
Memory regions in STM32F103C8

SRAM region : internal static RAM

- Stores global and local variables
- Contains the stacks, heaps and sometimes RTOS
- Volatile -> therefore flash is used to store variables permanently but all when necessary as it has limited number of writes

Peripheral region :

- To access on-chip peripherals GPIOs , I2C
- As mentioned above they're directed by the memory-map to control hardware devices



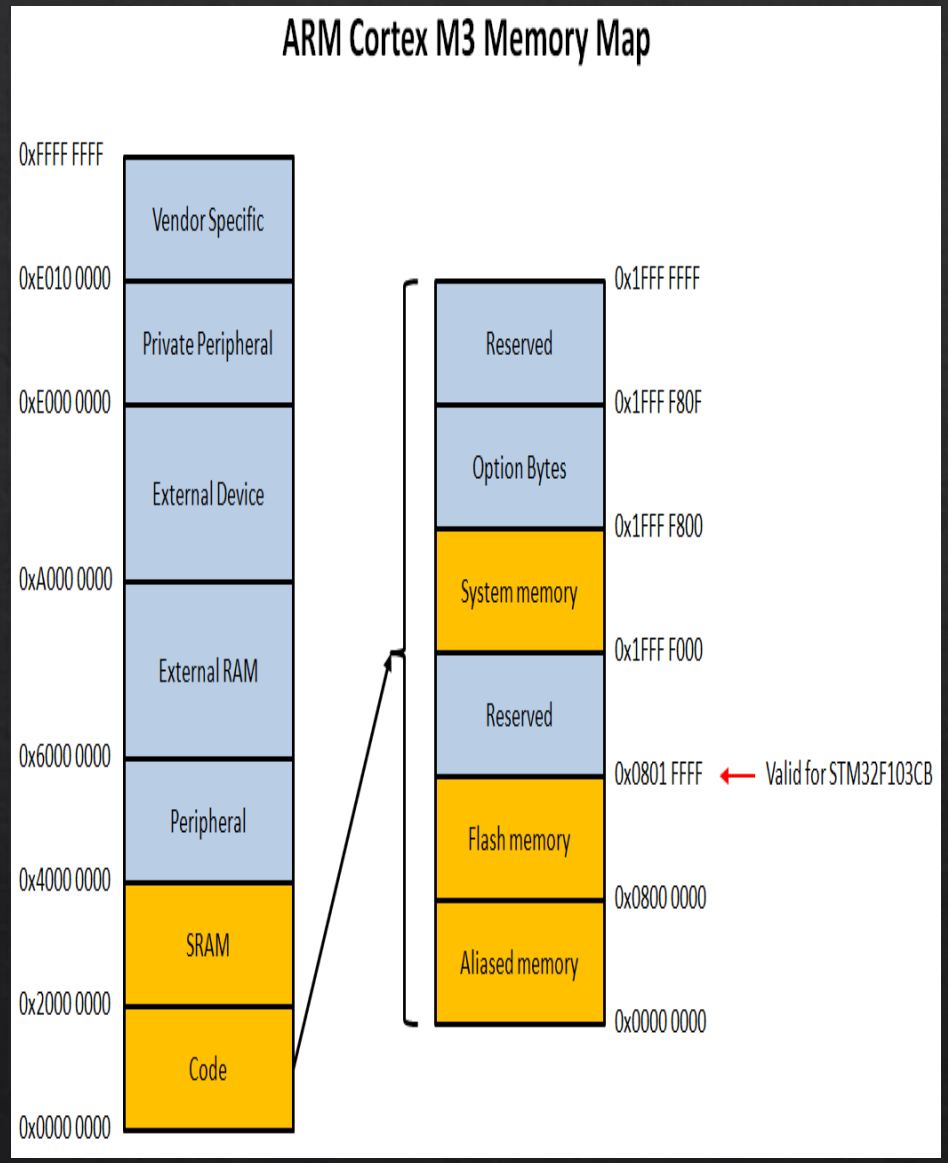
Memory regions in STM32F103C8

Option bytes :

- Controls some device settings, behaviors such as weather to boot from flash, system memory or SRAM
- -disables access to some flash sectors –can be done by manufacturer-

Private Peripheral

- Includes NVIC : interrupt control such as enabling/disabling
- -Systick timer : to control time delays , periodic interrupts



Base address

A **base address** is the **starting memory location** of a specific hardware block (like a peripheral or memory region).

- allows you to manually apply the correct offsets to reach other components - using the datasheet-.
- Used when **creating peripheral drivers**, since each driver depends on the correct base address to control the hardware.
- **Memory mapping** is built by assigning a base address to each peripheral
- With these addresses, you can **read/write hardware registers directly from software**, giving full control over some-parts that are usually hidden by manufacturer to protect the board from normal users

4.1 About the STM32 Cortex-M4 core peripherals

The address map of the *Private peripheral bus* (PPB) is:

Table 37. STM32 core peripheral register regions

Address	Core peripheral	Description
0xE000E010-0xE000E01F	System timer	*Table 55 on page 251*
0xE000E100-0xE000E4EF	Nested vectored interrupt controller	*Table 49 on page 219*
0xE000ED00-0xE000ED3F	System control block	*Table 53 on page 244*
0xE000ED88-0xE000ED8B	Floating point unit coprocessor access control	*Table 56 on page 252*
0xE000ED90-0xE000EDB8	Memory protection unit	*Table 44 on page 206*
0xE000EF00-0xE000EF03	Nested vectored interrupt controller	*Table 49 on page 219*
0xE000EF30-0xE000EF44	Floating point unit	*Table 56 on page 252*

References

https://nerdyelectronics.com/introduction-to-memory-mapping/#memory_and_peripheral_allocation

https://www.st.com/resource/en/datasheet/stm32f103c8.pdf

https://chatgpt.com/c/688b80b9-f494-8002-b432-8a12b5fd6855
--Chat gpt conversation--