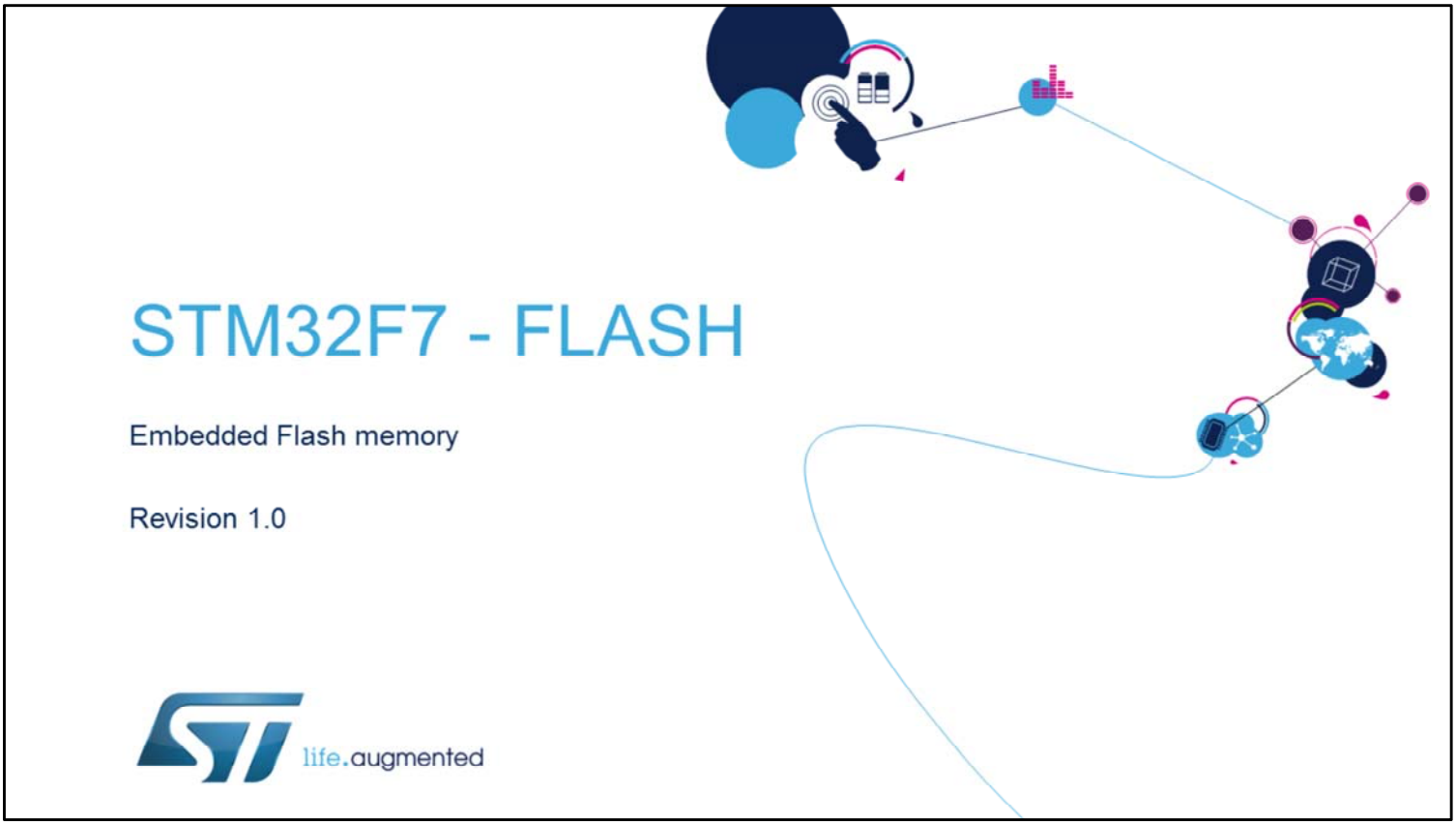
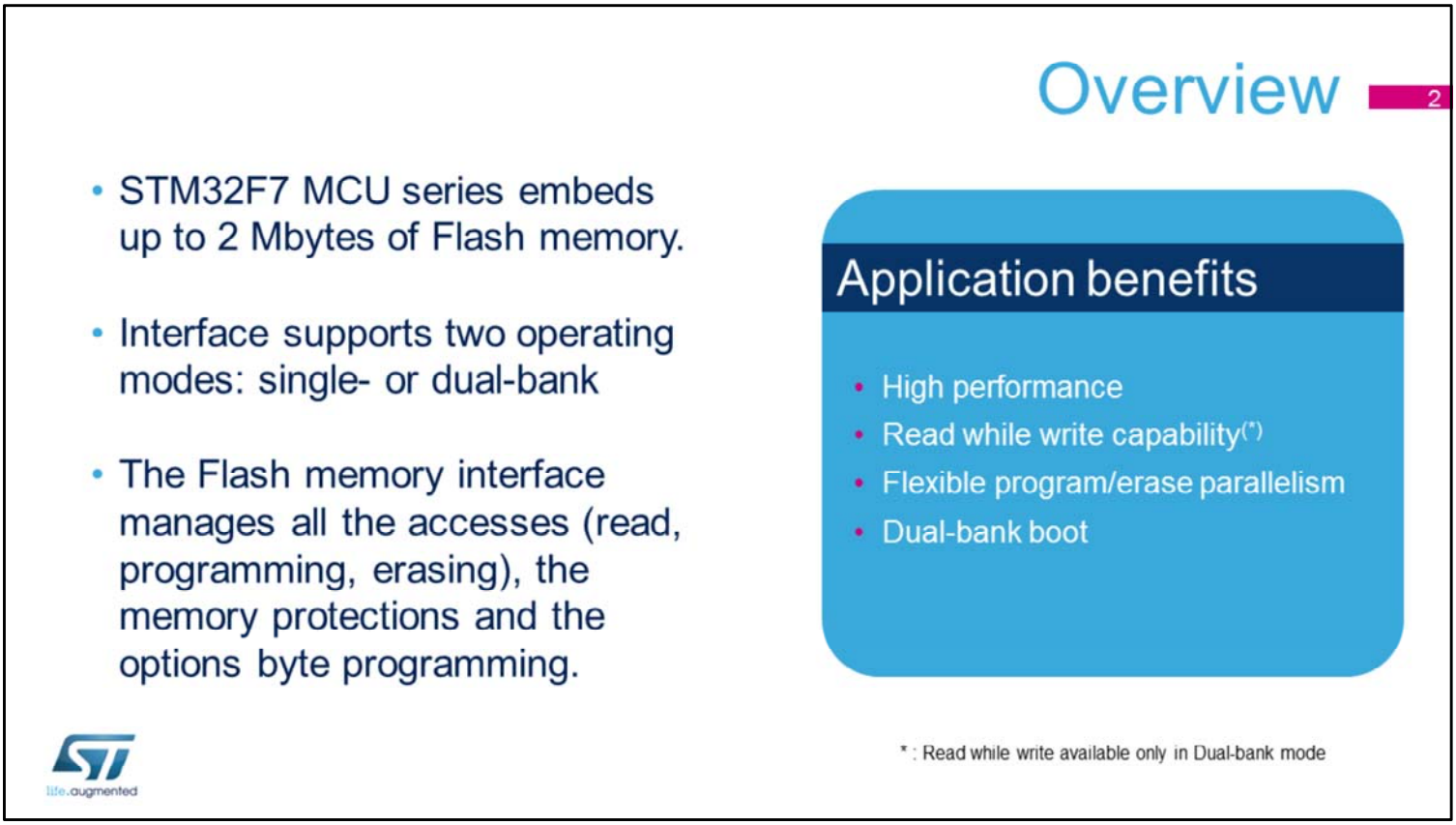
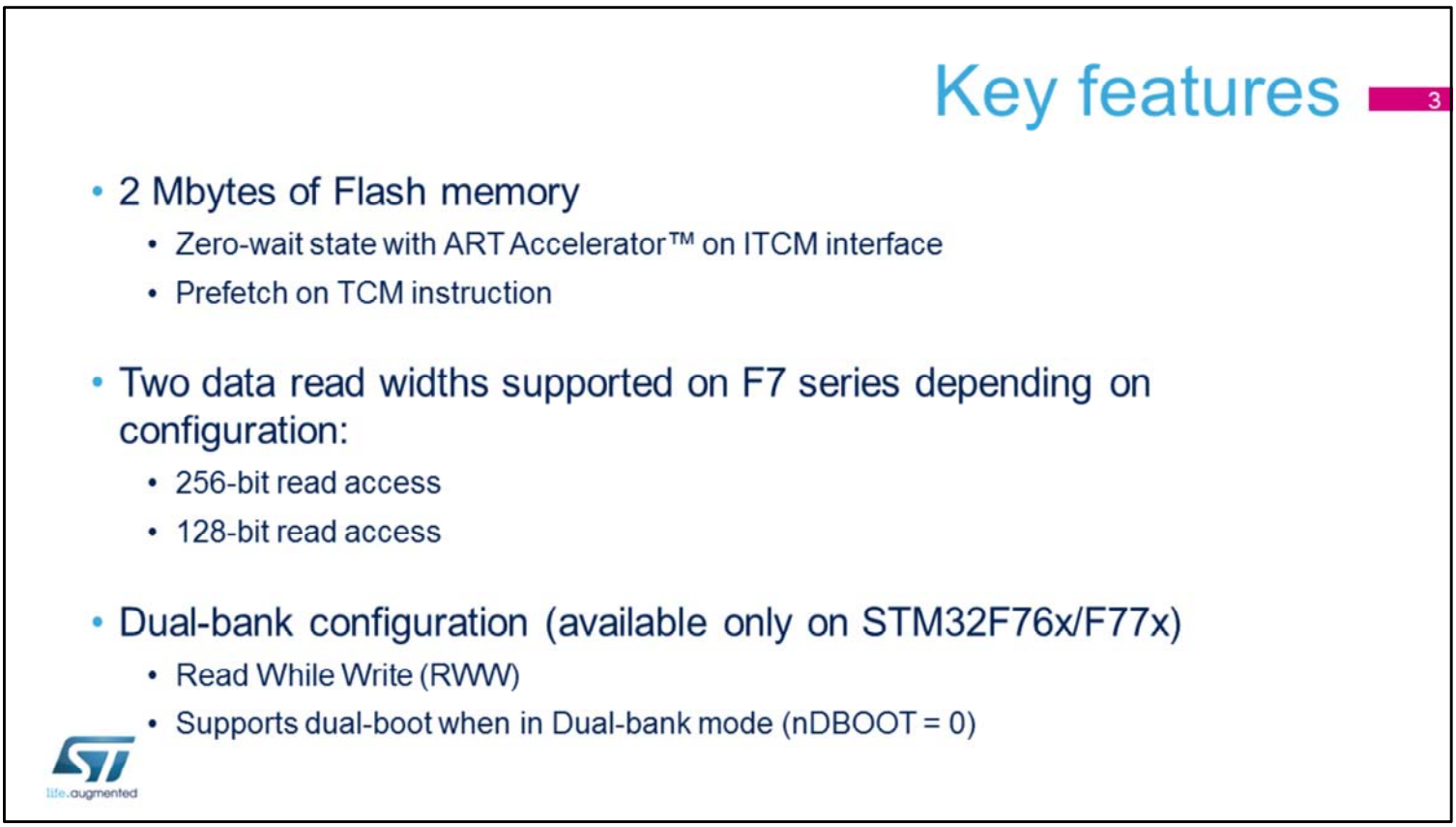


Hello, and welcome to this presentation of the STM32 Flash memory interface. It covers all the new features of the STM32F7 Flash memory.



STM32F7 microcontrollers embed up to 2 Mbytes of Flash memory.
The Flash memory interface manages all memory access (read, programming and erasing) as well as memory protection and option bytes.
Applications using this Flash memory interface benefit from its high performance together with low-power access. It supports read-while-write and allows dual-bank booting for devices supporting the dual-bank feature.



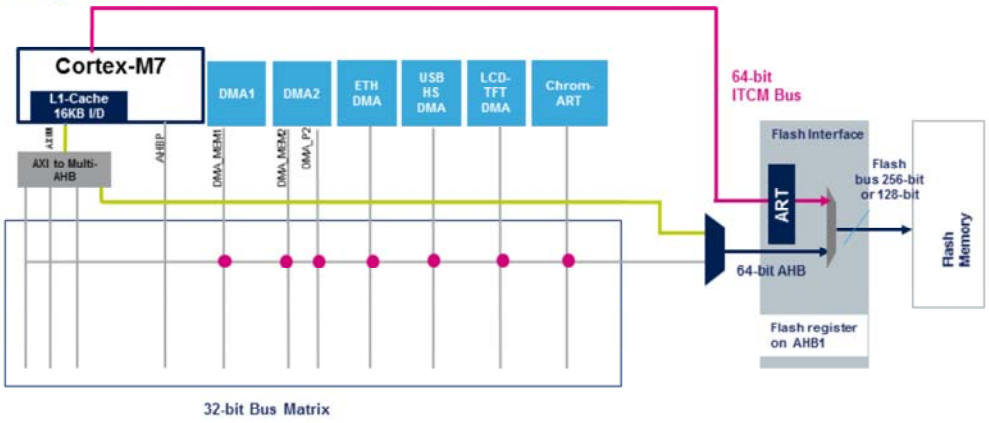
Flash memory has several key features.
It has the Adaptive Real-Time (ART) memory accelerator, with an unified instruction and data cache also a prefetch buffer allows a linear performance in relation to frequency while executing from the tightly-coupled memories for instructions (ITCM) interface.

The STM32F7 series has up to 2 Mbytes of Flash memory configurable in Single- or Dual-bank mode, with a read-while-write capability that can program or erase one bank while executing code from the other bank.

Note that the dual-bank feature is only available on STM32F76x/F77x devices.

Flash memory interfaces

- 3 Main interfaces
  - 64-bit ITCM interface
  - 64-bit AXI/AHB interface
  - 32-bit AHB register interface



life.augmented

The Flash memory interface has three main buses, a 64-bit ITCM interface, a 64-bit AXI/AHB interface and a 32-bit AHB register interface.

Flash interfaces

- 64-bit ITCM interface:
  - Connected to the Cortex-M7 ITCM bus and used for code execution and data read accesses.
  - Write accesses are not supported on ITCM interface
- 64-bit AHB interface:
  - Connected to the Cortex-M7 AXI bus through the AHB bus matrix and used for code execution, data read and write accesses.
  - DMA controllers and peripherals DMA data transfers on the Flash memory are supported only through the AHB interface.
- 32-bit AHB register interface:
  - Used for control and status register accesses

life.augmented

The Flash memory is mapped on the ITCM interface of the Cortex-M7 processor and used for code execution and data read accesses to the Flash memory. Write accesses to the Flash memory are not possible through this interface.
The Flash memory remains accessible to the Cortex-M7 processor and other masters such as DMA controllers through the AHB interface.
The 32-bit AHB register interface is used for control and status register accesses during user Flash memory and option byte programming.

Flash organization

- Flash memory is organized as follows:

Main Flash memory	When option byte
• Single-bank mode (Default mode)	nDBANK = 1
• Dual-bank mode (only on STM32F76x/F77x)	nDBANK = 0
Information block	
• 60 Kbytes of System memory from which the device boots in System memory boot mode	
• 1024 Bytes OTP: one-time programmable	
• **User Option bytes** for user configuration	

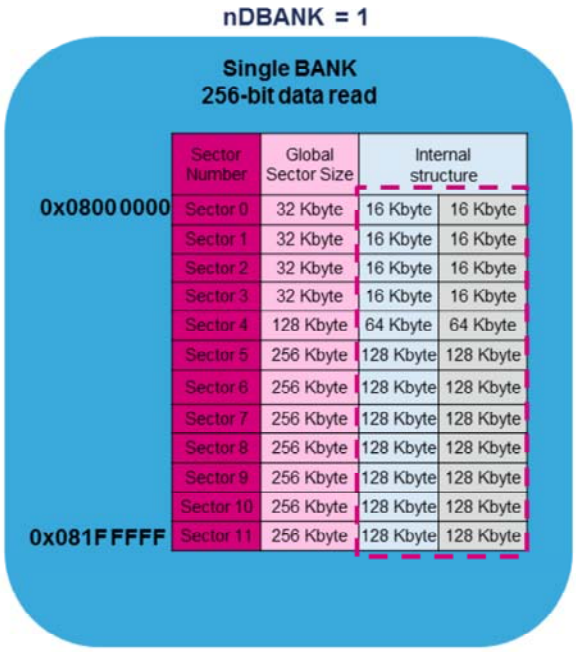
The Flash memory is divided into 2 functional blocks, a main memory block containing up to 2 Mbytes for user code and data. The default configuration for the Main Flash memory is in Single-bank mode. For devices supporting Dual-bank mode, the Flash memory organization can be changed via the nDBANK user option bit.

The second Flash memory functional block is an information block which contains 3 parts.
The first part is the system memory which is reserved for use by STMicroelectronics and contains the bootloader. When selected, the device boots in System memory to execute the bootloader.

The second part is a 1-Kbyte one-time programmable area. The OTP area cannot be erased and can be

written to only once.
The last part contains the option bytes for configuring user options.

Flash organization: 2 MB single bank

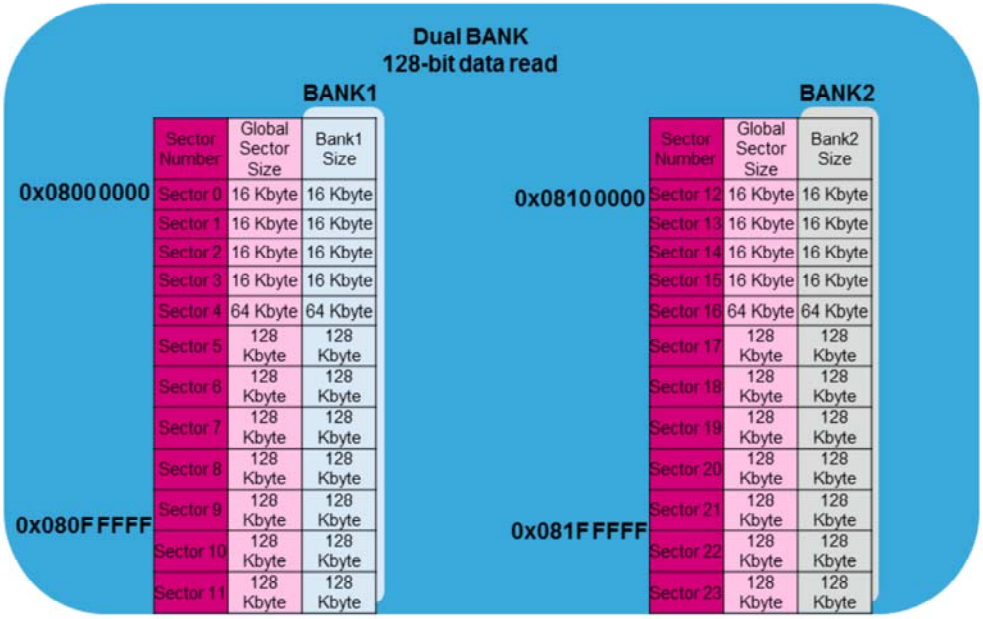


This slide shows an example of the Flash memory map. This example is from STM32F76x/F77x devices with 2 Mbytes of Flash memory.
In Single-bank mode, there are 4 sectors of 32 Kbytes starting from sector 0 to 3, one sector of 128 Kbytes which is sector 4 and seven sectors of 256 Kbytes starting from sector 5 to sector 11.
The sector number is used in the software procedure to erase a sector or to protect it.

Flash organization: 2 MB Dual bank

SWP_FB = 0

User option nDBANK = 0

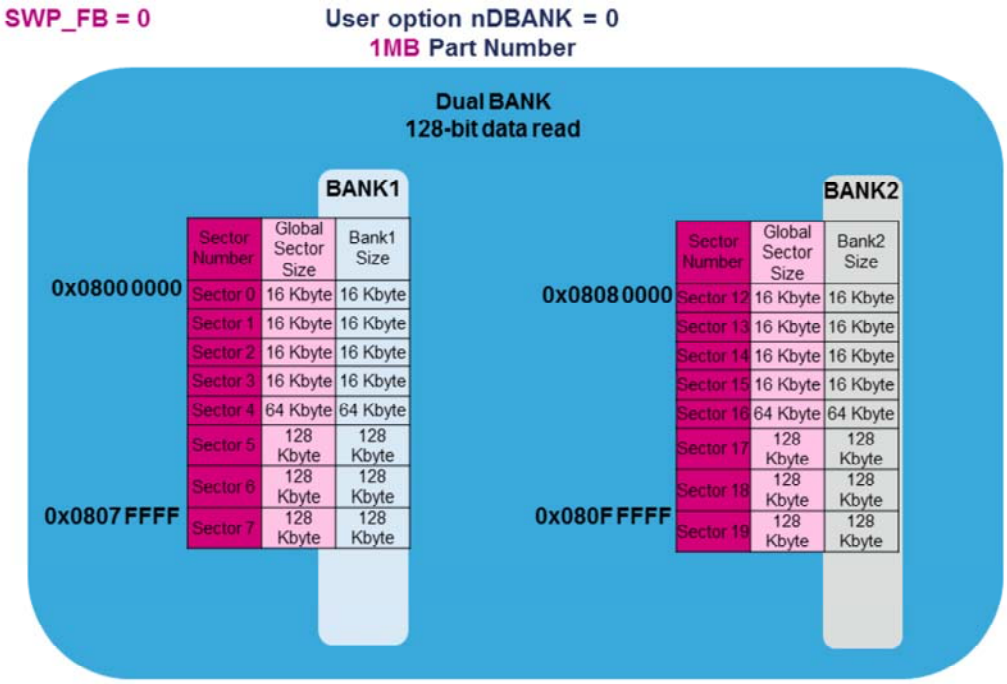


Sector Number	Global Sector Size	Bank1 Size
Sector 0	16 Kbyte	16 Kbyte
Sector 1	16 Kbyte	16 Kbyte
Sector 2	16 Kbyte	16 Kbyte
Sector 3	16 Kbyte	16 Kbyte
Sector 4	64 Kbyte	64 Kbyte
Sector 5	128 Kbyte	128 Kbyte
Sector 6	128 Kbyte	128 Kbyte
Sector 7	128 Kbyte	128 Kbyte
Sector 8	128 Kbyte	128 Kbyte
Sector 9	128 Kbyte	128 Kbyte
Sector 10	128 Kbyte	128 Kbyte
Sector 11	128 Kbyte	128 Kbyte

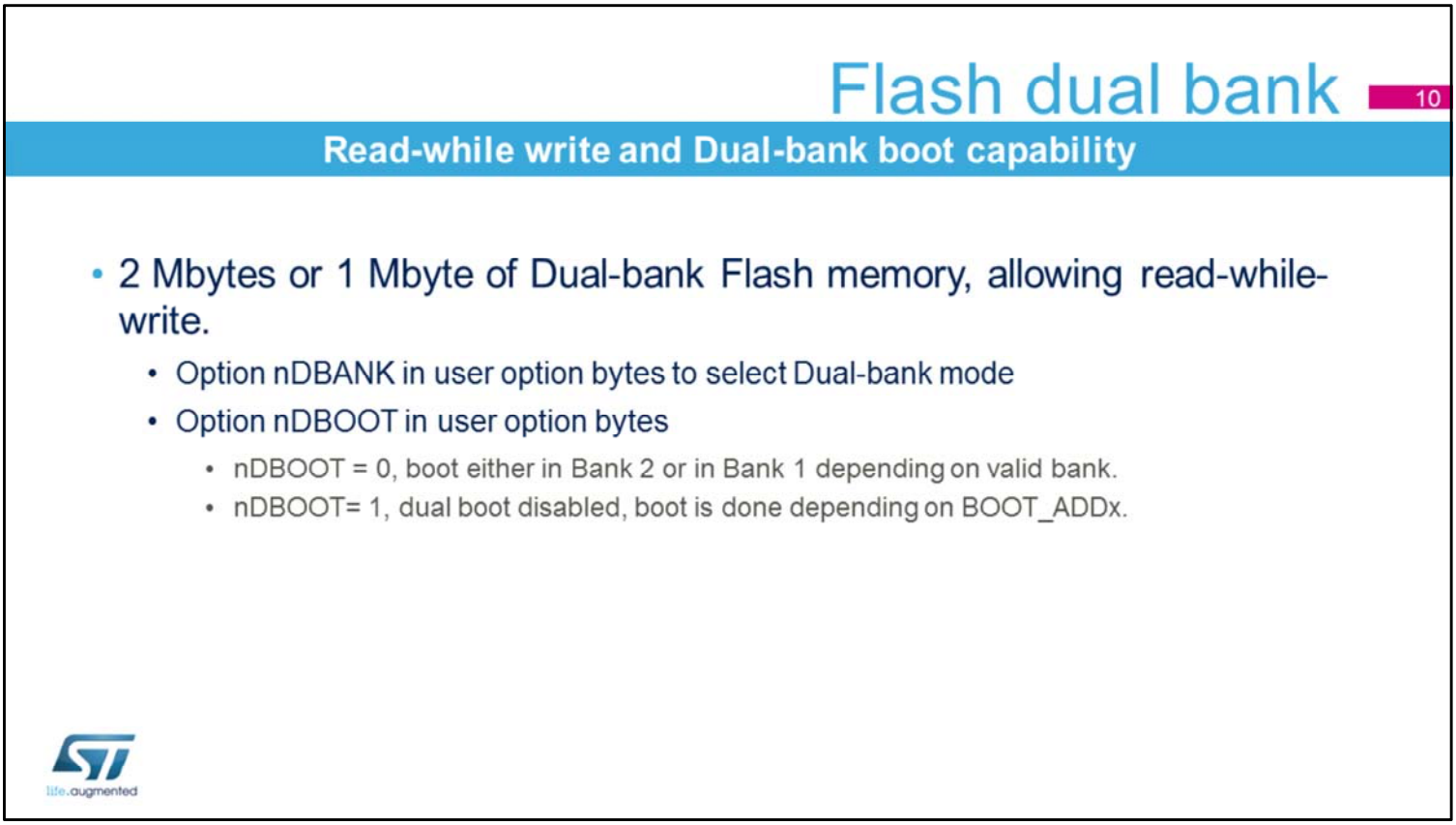
Sector Number	Global Sector Size	Bank2 Size
Sector 12	16 Kbyte	16 Kbyte
Sector 13	16 Kbyte	16 Kbyte
Sector 14	16 Kbyte	16 Kbyte
Sector 15	16 Kbyte	16 Kbyte
Sector 16	64 Kbyte	64 Kbyte
Sector 17	128 Kbyte	128 Kbyte
Sector 18	128 Kbyte	128 Kbyte
Sector 19	128 Kbyte	128 Kbyte
Sector 20	128 Kbyte	128 Kbyte
Sector 21	128 Kbyte	128 Kbyte
Sector 22	128 Kbyte	128 Kbyte
Sector 23	128 Kbyte	128 Kbyte

In Dual-bank mode (for STM32F76x/F77x MCUs), the 2 Mbytes of Flash memory is organized as 12 sectors for each Flash memory bank. Where each bank has 4 sectors of 16 Kbytes, one sector of 64 Kbytes and 7 sectors of 128 Kbytes. Bank 2 sectors start from sector 12 to sector 23.

Flash organization: 1 MB dual bank



Dual-bank mode is also available for STM32F76x/F77x MCUs that are 1-Mbyte devices. This slide describes the memory organization in this configuration.



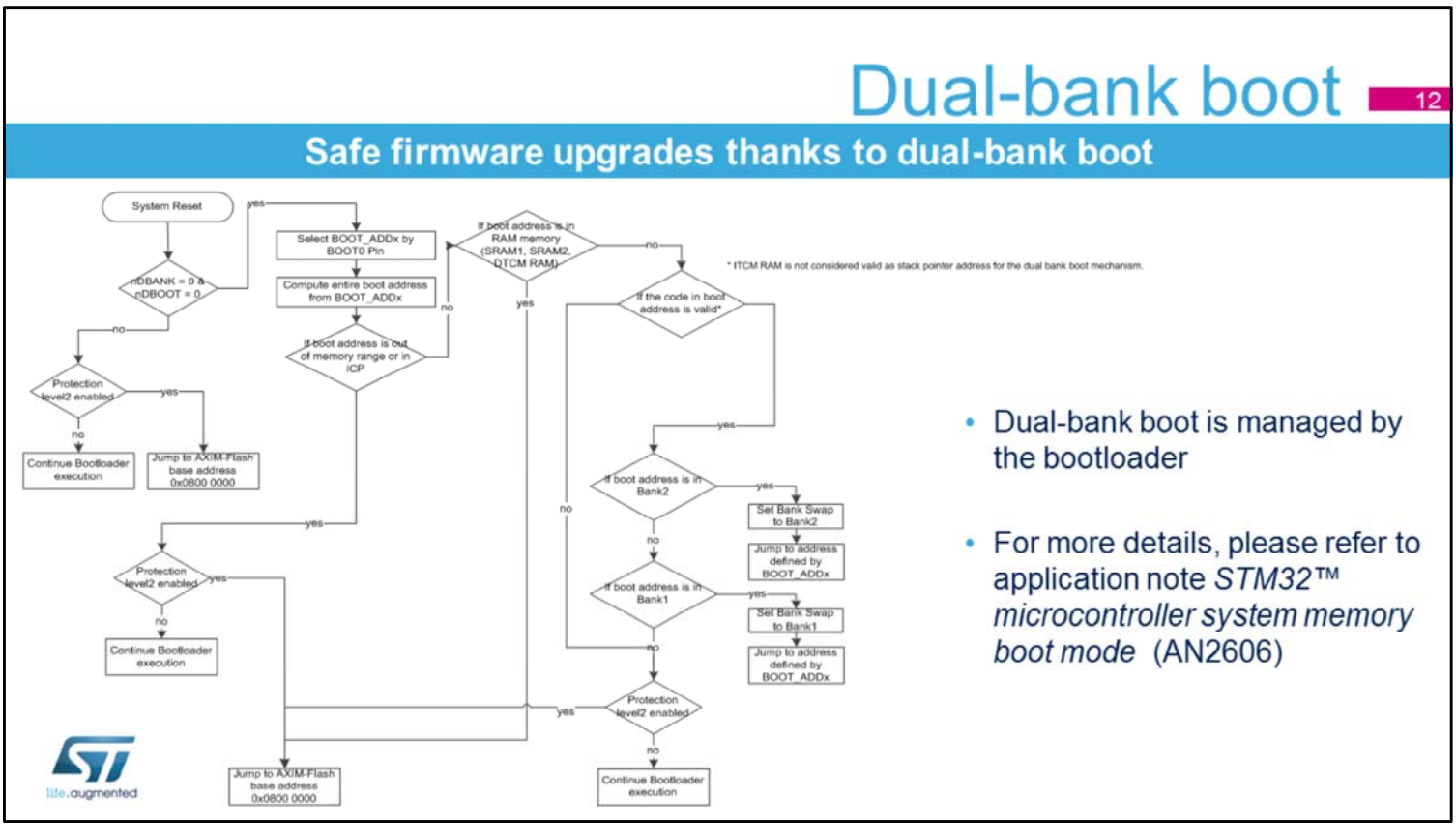
The embedded Flash memory is a dual-bank memory with read-while-write and dual-bank boot capability, able to boot either from Bank 1 or Bank 2.
The nDBOOT option in the user option bytes is used to select the dual-bank boot mode. When the nDBOOT option is reset, the device boots either from Bank 2 or from Bank 1 depending on the BOOT_ADDx address and valid bank content.
When dual-bank boot is enabled, the device boots from System memory to execute bank swap when valid bank is detected.

Flash dual bank

- Read-while-write
  - Thanks to dual bank feature, it is possible to read from one bank while programming/erasing the other bank
  - When programming/erasing is done in the same bank, the CPU access is stalled as long as BUSY = 1
- Safe Firmware upgrade thanks to dual boot
  - Dual-bank boot is managed by the bootloader firmware
  - Dual-bank boot is implemented via the SWP_FB bit in SYSCFG_MEMRMP

life.augmented

With a dual-bank memory, it is possible to read from one bank while programming or erasing the other bank. Code execution is not stopped when the Flash memory is being programmed. When programming or erasing data in the same bank, the fetch is stalled as long as the Flash memory controller is busy.
Using the FB_MODE bit in the System Configuration memory remap register, the two Flash memory bank addresses can be swapped. When this bit is cleared, Bank 1 is mapped at address 0x0800 0000 for AHB access and 0x00200000 for ITCM access. When this bit is set, Bank 2 is mapped at address 0x0800 0000 for AHB access and 0x00200000 for ITCM access which allows the device to boot into Bank 2.



The dual-bank boot allows a safe firmware upgrade as the previous firmware version is still present in the other memory bank. The dual-bank boot is managed by the bootloader. The device boots in Bank 2 using the nDBOOT option bit, programmed in the Flash option bytes.

The boot pin and boot option are configured for booting in Flash memory. If the nDBOOT option bit is set, the device boots according to address in BOOT_ADDx option bytes. If the nDBOOT option bit is cleared, the device boots in the System Flash memory. The bootloader checks the bank's content indexed by BOOT_ADDx address, as it must read there the stack pointer at that location. If Bank 2's BOOT_ADDx address content is a valid SRAM address, the bootloader swaps the banks to remap the Bank 2 at address 0x00200000

and jumps into Bank 2. If it is not valid, the bootloader swaps the banks to remap Bank 1 at address 0x00200000 and jumps into Bank 1.
Note that the bootloader uses resources in SRAM for dual-boot, the content of SRAM is not preserved after reset.

Flash memory features

Flash Program/Erase operations

- Flash program/Erase operations
  - Programming granularity: Byte, half-word, word and double-word
  - Erasing granularity: Sector Erase, Bank Erase and Mass Erase
- The maximum program/erase parallelism is limited by the supply voltage and by whether the external Vpp supply is used or not.

	Voltage range 2.7 - 3.6 V with External VPP	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.8V - 2.1 V
Maximum Parallelism	x64	x32	x16		x8
PSIZE[1:0]	11	10	01		00

The Flash memory interface implements program and erase operations. The programming granularity depends on the MCU supply range. Word parallelism is only available in the 2.7 to 3.6 V range. Half-word parallelism is available starting from 2.1 to 2.7 V while byte programming is available on the full VDD range. To enhance programming parallelism, it is possible to use double-word programming, this is only available starting from 2.7 V while applying an external Vpp while programming.

Erase granularity can be down to sector size, also it is possible to erase the complete Flash memory using the Mass Erase option. Bank Erase is only available with devices supporting Dual-bank mode.

Flash memory read access

14

462 DMIPS at 216 MHz

HCLK clock frequency (MHz)

Wait States (WS) (LATENCY)	Voltage range 2.7 V - 3.6 V	Voltage range 2.4 V - 2.7 V	Voltage range 2.1 V - 2.4 V	Voltage range 1.8V - 2.1 V
0WS(1CPU cycle)	0 < HCLK <= 30	0 < HCLK <= 24	0 < HCLK <= 22	0 < HCLK <= 20
1WS(2CPU cycle)	30 < HCLK <= 60	24 < HCLK <= 48	22 < HCLK <= 44	20 < HCLK <= 40
2WS(3CPU cycle)	60 < HCLK <= 90	48 < HCLK <= 72	44 < HCLK <= 66	40 < HCLK <= 60
3WS(4CPU cycle)	90 < HCLK <= 120	72 < HCLK <= 96	66 < HCLK <= 88	60 < HCLK <= 80
4WS(5CPU cycle)	120 < HCLK <= 150	96 < HCLK <= 120	88 < HCLK <= 110	80 < HCLK <= 100
5WS(6CPU cycle)	150 < HCLK <= 180	120 < HCLK <= 144	110 < HCLK <= 132	100 < HCLK <= 120
6WS(7CPU cycle)	180 < HCLK <= 210	144 < HCLK <= 168	132 < HCLK <= 154	120 < HCLK <= 140
7WS(8CPU cycle)	**210 < HCLK <= 216**	168< HCLK <= 192	154 < HCLK <= 176	140 < HCLK <= 160
8WS(9CPU cycle)	-	**192 < HCLK <= 216**	176 < HCLK <= 198	**160 < HCLK <= 180**
9WS(10CPU cycle)	-	-	**198 < HCLK <= 216**	-

216 MHz is achieveable only when Over-drive mode is active

life.augmented

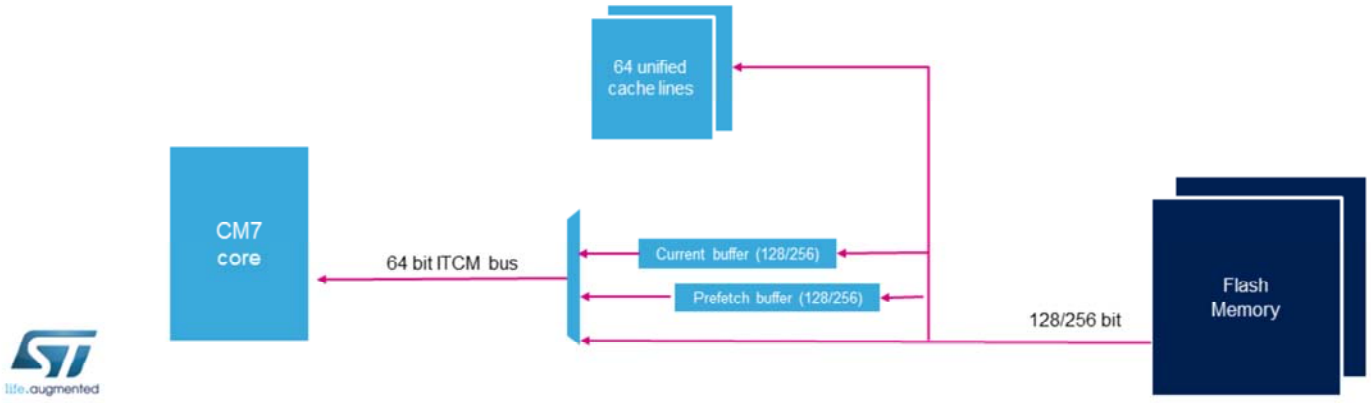
As the Flash memory has a long read access time, in order to read the Flash memory, software is required to configure the number of wait states to be inserted in a read access, depending on the clock frequency.
The number of wait states also depends on the MCU supply voltage range.
For the 2.7 to 3.6 V range, the Flash memory can be accessed at up to 216 MHz with 7 wait states. It can be accessed with 0 wait states at up to 30 MHz.
For the 2.4 to 2.7 V range, code execution at up to 216 MHz is possible with 8 wait states.
For the 2.1 to 2.4 V range, code execution at up to 216 MHz is possible with 9 wait states
When VDD is below 2.1 V, Flash memory can be accessed with 0 wait state at up to 20 MHz.
Thanks to the Cortex-M7 L1 cache and to the adaptive real-time memory accelerator, the ART accelerator on

the ITCM interface, the program can be executed with 0 wait states independent of the clock frequency. This provides an almost linear performance in relation to frequency.

Adaptive real-time memory accelerator (ART Accelerator™)

Outstanding performances

- ART Accelerator™ on **ITCM interface**
  - 64 unified cache lines of 256 bits or 128 bits depending on devices and configuration (in Single- or Dual-bank mode)
  - **Prefetch buffer**



The ART accelerator is available only on the Cortex-M7 ITCM interface. It brings outstanding performance and reduces cache usage while executing code from Flash memory. It consists of 64 unified cache lines with a 256- or 128-bit width depending on the Flash memory interface width configuration. For each Cortex-M7 fetch a lookup in current buffer, prefetch buffer and ART cache is performed . If there is a hit, the data is immediately returned to the CPU.

In case of a miss, the Cortex M7 processor takes the data directly from the Flash memory. In parallel, the Flash memory line is copied into the current buffer enabled and ART Cache if enabled.

If prefetch is enabled, another Flash memory access is performed to fill the prefetch buffer with sequential data.

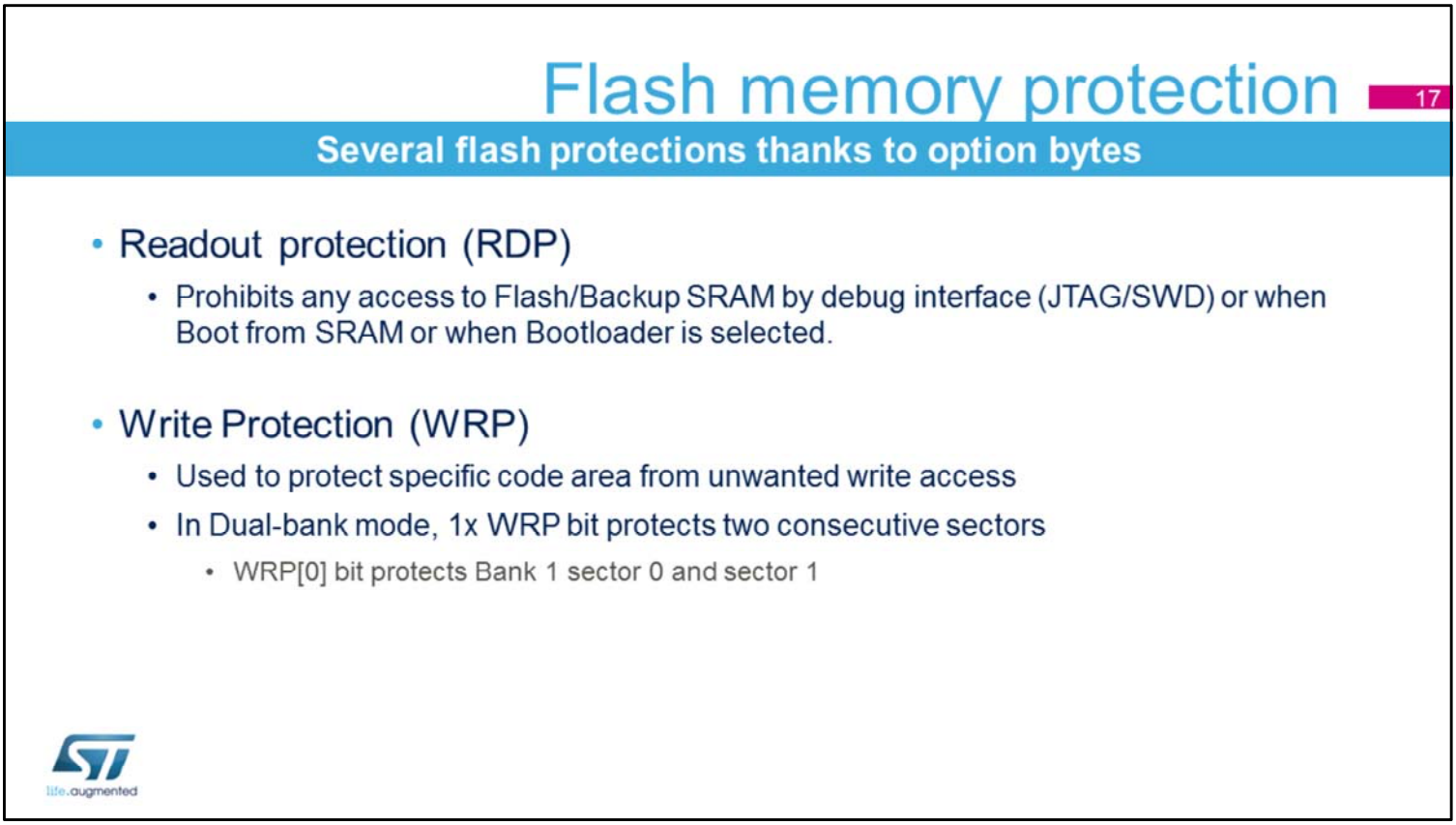
User option bytes

Options	Description
BOR_LEV[1:0]	BOR reset Level
nRST_STOP; nRST_STDBY;	Reset/No Reset generated when entering the Stop/Standby mode
WWDG_SW; IDWG_SW **IWDG_STOP; IWDG_STDBY**	Hardware/Software window watchdog/ independent watchdog Independent watchdog counter is frozen/not frozen in Stop/Standby mode
nDBOOT **nDBANK**	Dual-bank boot Dual-bank mode activation for Flash memory
BOOT_ADDx x=0,1	Boot address configuration
nWRP[xx:0]	Sector write protection activation
RDP[7:0]	Read protection level

- The user options bytes are loaded:
  - After a Power reset (BOR or exit from Standby)
  - When OPTSTRT bit is set in the Flash control register (FLASH_OPTCR).

Several option bytes are available in the Flash memory information block to configure certain specific features of the device. The option bytes are saved in the Flash memory after programming.
The user option bytes are loaded in two cases: either after a power or a brown-out reset when exiting from Standby mode, or after option byte programming.
Three option bits are used to configure the brown-out reset threshold.
Two options are available to prohibit or allow the Stop and Standby low-power modes.
Four options are available to configure whether the watchdogs are enabled by hardware or after a software configuration, and if the independent watchdog is frozen or not in Stop and Standby modes.
Two options are used to enable dual-bank boot mode and dual-bank boot in devices supporting Dual-bank

mode.
The BOOT_ADDx options are used together with the BOOT0 pin to configure the memory address used for booting.
The RDP option byte is used to set the memory readout protection level.



Several Flash memory protection options can be configured using the option bytes.
The readout protection is configured using the RDP option byte. The readout protection prohibits any access to the Flash memory or the backup registers by the debug interface or when booting from SRAM or when the bootloader is selected.

The write protection is configured using the WRP option byte. This option protects specific code areas from unwanted write accesses. The write-protected area can be defined with sector granularity for single-bank Flash memory devices while in Dual-bank mode, one bit protects two consecutive sectors.

Interrupts

Interrupt event	Description
End of operation	Set by hardware when one or more Flash memory operation (programming / erase) has been completed successfully.
Programming error	Set by hardware when a Flash memory operation (program / erase) completes unsuccessfully.
Write protection error	Set by hardware when attempt to program or erase in a write protected area, System memory area or OTP area

Three interrupts can be generated by the Flash memory.
The end-of-operation interrupt is triggered when one or more Flash program or erase operations is completed successfully.
The programming error interrupt is triggered when a Flash memory program or erase operation failed.
The write protection error interrupt is triggered when a write access is attempted to a write-protected area of the Flash memory.

Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop	Flash clock off. Peripheral registers content is kept. Flash memory can be configured in Power-down mode during Stop
Standby	Powered-down. The Flash memory interface must be reinitialized after exiting Standby mode.

This shows a summary of the Flash memory interface states in each of the power modes .

Flash memory performance

5 CoreMark / MHz

Flash memory performance is almost linear with frequency.
1082 CoreMark at 216 MHz (7WS) = ~5 **CoreMark / MHz** (Caches ON, Prefetch ON)

Execution setup	STM32F7xx CoreMark / MHz	STM32F76x Single bank CoreMark / MHz	STM32F76x Dual bank CoreMark / MHz
Code execution from Flash over ITCM interface (ART and prefetch ON)	5,01	5,01	4,93
Code execution from Flash over AXIM interface (I-cache + D-cache ON)	4,98	5,04	5,04
Code execution from TCM interface (code in ITCM RAM and data in DTCM RAM)	5,04	5,04	5,04

Toolchain	MDK-ARM version : 5.16.0.0

life.augmented

The performance of the Flash memory is almost linear with the frequency using the ART accelerator or Cortex-M7 processor using the Instruction and Data cache.
The CoreMark score is 1082 at 216 MHz, which corresponds to ~5 CoreMark / MHz when executing from the ITCM interface with the Instruction Cache, Data Cache and Prefetch buffer enabled or when executing over the AXIM interface with the Cortex-M7 L1 cache enabled.

Related peripherals

- Refer to these trainings linked to this peripheral
  - System configuration controller (SYSCFG)
  - Reset and clock controller (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC)

life.augmented

This is a list of peripherals related to the Flash memory interface. Please refer to these trainings for more information if needed.

References

- For more details, please refer to the following documents:
  - STM32 microcontroller system memory boot mode (AN2606)
  - STM32F7 Series Flash memory dual bank mode (AN4826)
  - STM32F7 Series system architecture and performance (AN4667)

life.augmented

For more details, please refer to these application notes about the STM32F7 microcontroller execution performances, dual-bank feature and system memory boot mode.