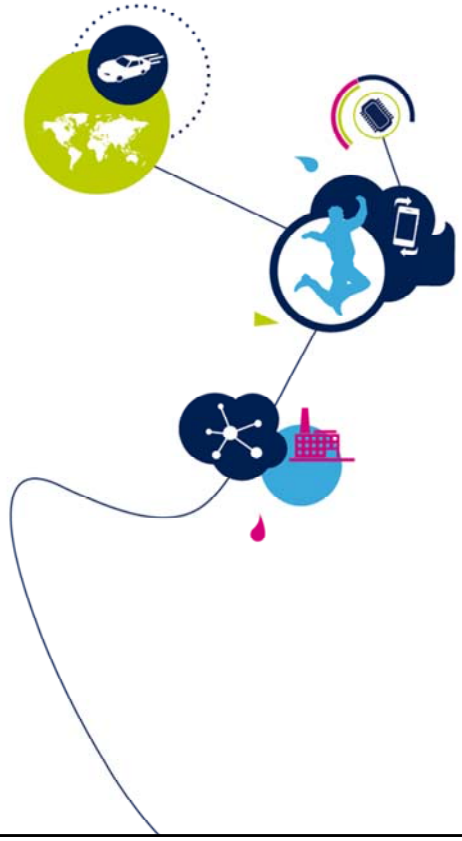
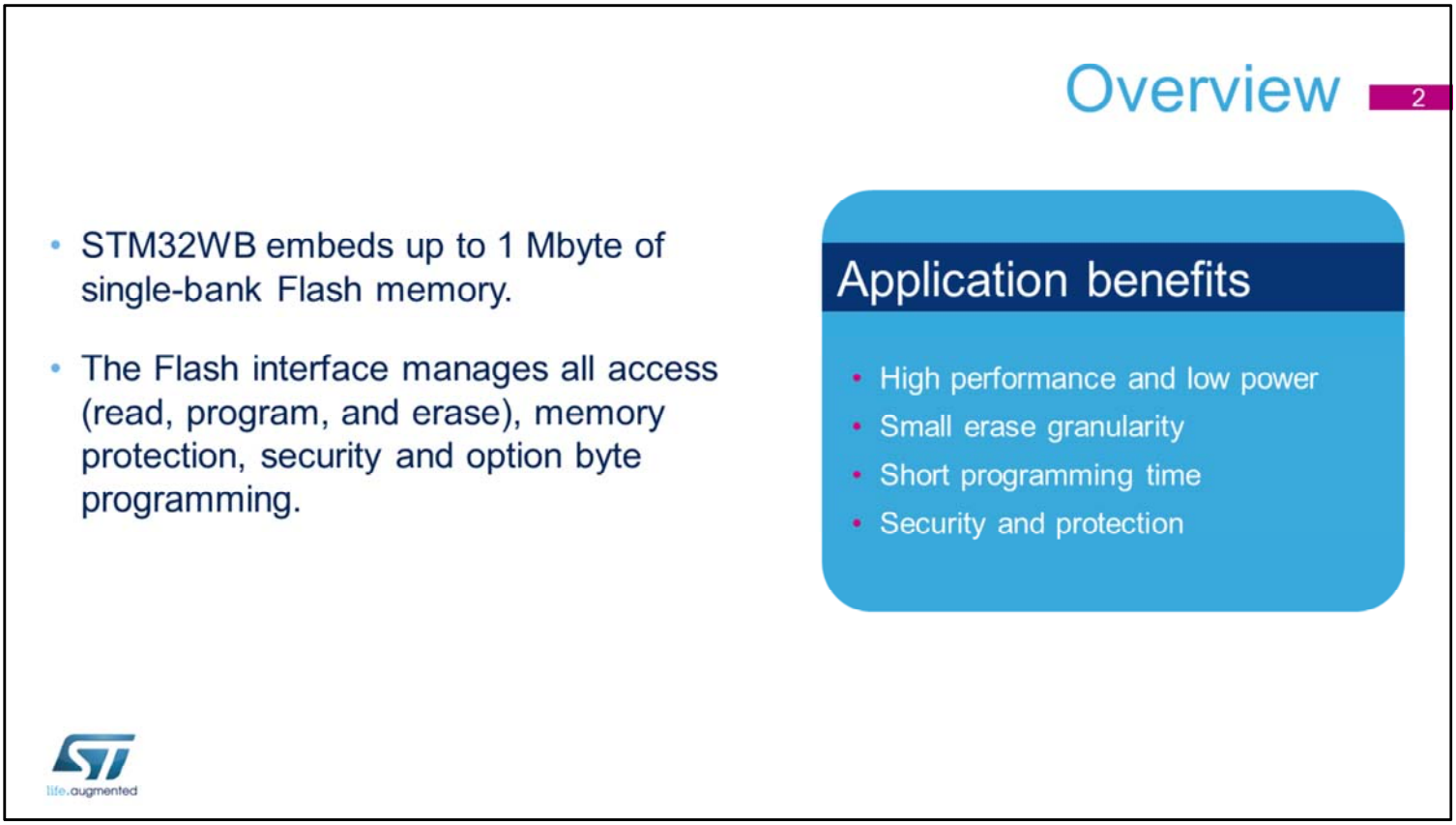
STM32WB - FLASH

Embedded Flash memory
Revision 1.0

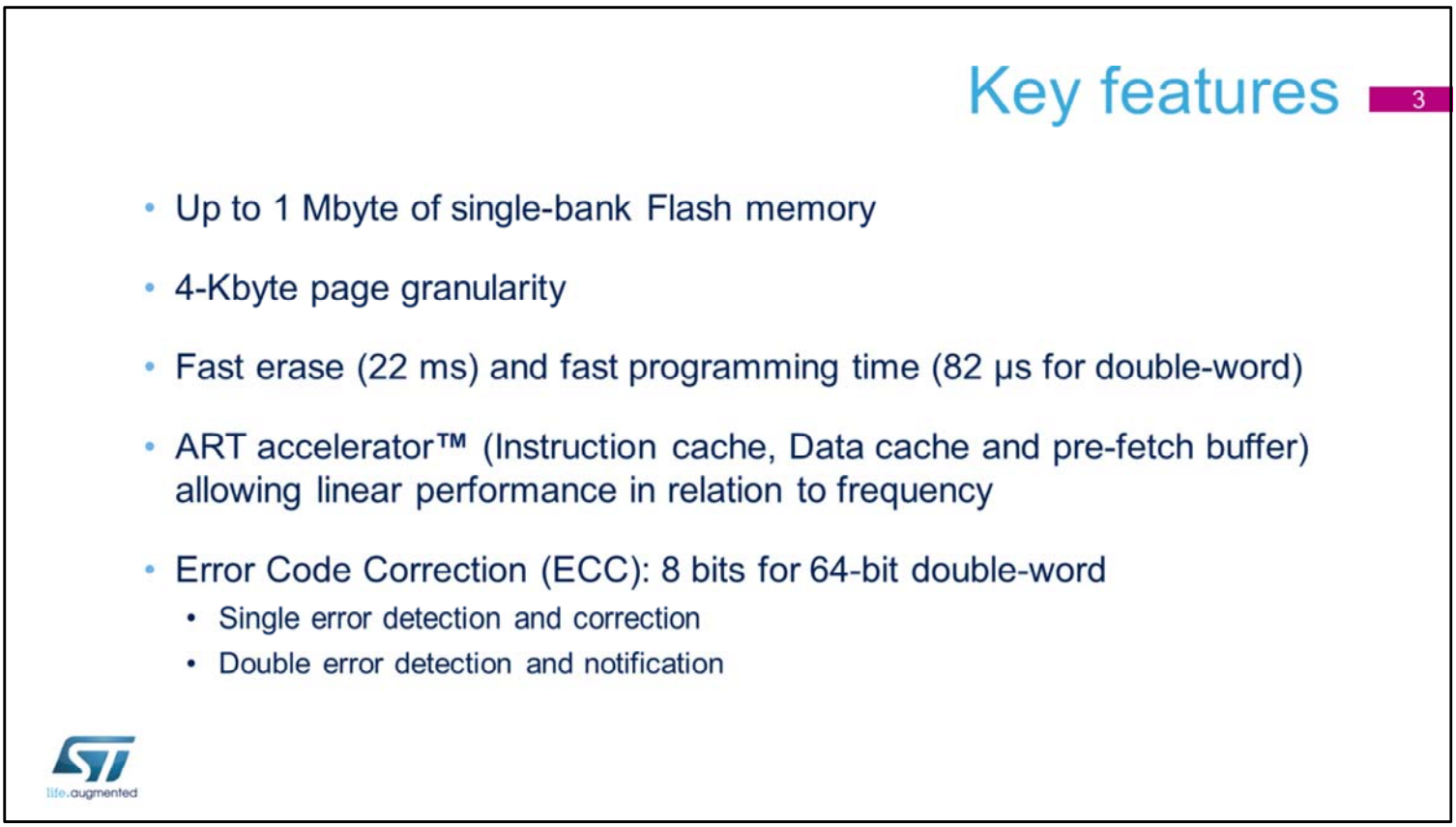


life.augmented

Hello, and welcome to this presentation of the STM32WB Flash memory. All STM32WB Flash memory features will be presented.



The STM32WB embeds up to 1 Mbyte of single-bank Flash memory.
The Flash memory interface manages all memory access (read, program and erase) as well as memory protection, security and option bytes.
Applications using this Flash memory interface benefit from its high performance together with low-power access. It has a small erase granularity and short programming time.
The STM32WB Flash memory provides various security and protection mechanisms for code and data, read and write access.



The STM32WB's Flash memory has several key features.
It has up to 1 Mbyte of single-bank Flash memory.
The erase granularity, corresponding to the page size, is only 4 Kbytes.
A page, bank or mass erase operation requires only 22 ms, and the programming time is only 82 µs for a double-word.
The adaptive real-time memory accelerator, with an instruction cache, a data cache and a pre-fetch buffer, allows a linear performance in relation to frequency.
The Flash memory supports Error Code Correction (ECC) which is 8 bits long for each 64-bit double word. A single error is detected and corrected. A double error is detected, but not corrected.

Flash memory organization

The Flash memory is organized as follows:

- A Main memory block containing 256 pages of 4 Kbytes each. Each page is made of 8 rows of 512 bytes.
- An Information block containing:
  - System memory reserved for ST bootloader.
  - OTP (one-time programmable) 1-Kbyte (128 double-words) area for user data.
    - Data in the OTP area can't be erased and a double-word can be written only once. If only one bit is set to '0', the entire double-word can no longer be written, except with a value of all 0x0.
  - Option bytes for user configuration.

life.augmented

The Flash memory contains 256 pages of 4 Kbytes each. Each page is made of 8 rows of 512 bytes.
Next to the main memory block, there is an information block which contains 3 parts.
The first part is the system memory which is reserved for the STMicroelectronics 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 a double-word can be written only once. If one double-word bit is at '0', the entire double-word can no longer be written, except with the value all zeros. Programming a previously programmed double-word is only allowed when programming all zeros.
The last part contains the option bytes for configuring user options.

Flash memory organization

Flash area	Flash memory address	Size	Name
Main memory	0x0800 0000– 0x0800 0FFF	4 Kbytes	Page 0
	…	…	…
	0x080F F000– 0x080F FFFF	4 Kbytes	Page 255
Information block	0x1FFF 0000– 0x1FFF 6FFF	28 Kbytes	System memory
	0x1FFF 7000– 0x1FFF 73FF	1 Kbyte	OTP area
	0x1FFF 8000– 0x1FFF 807F	128 bytes	Option bytes

This slide shows the Flash memory map. There are 256 pages for the main memory, starting from page 0. The page number is used in the software procedure to erase a page.

Robust memory integrity and safety

- ECC (Error Code Correction): 8 bits long for a 64-bit word
  - Single error correction: ECCC bit set in FLASH_ECCR, optional interrupt generation
  - Double error detection: ECCD bit set in FLASH_ECCR => NMI
  - Failure address saved in FLASH_ECCR register
- Programming granularity is 64 bits (really 72 bits incl. 8-bit ECC)
  - 2 programming modes :
    - Standard (for main memory and OTP)
    - Fast (main memory only). **Programs 64 double-words without verifying the Flash location.**

life.augmented

The Flash memory embeds an Error Code Correction function to ensure robust memory integrity and safety.
The ECC is 8 bits long for a 64-bit word. In case of a single error, it is corrected. The ECCC bit is set in the Flash ECC register, and an interrupt is generated if it is enabled. In case of a double error, it is detected but not corrected. The ECCD bit is set in the Flash ECC register, and a non-maskable interrupt is generated. When an ECC error is detected, the failure address is saved in the Flash ECC register.
The programming granularity is 64 bits, in fact it's 72 bits with the 8-bit ECC. There are 2 programming modes: Standard mode for the main memory and OTP, and Fast mode for the main memory only. In Standard mode, the Flash memory checks that the double-word is erased before launching the programming. In Fast mode, 64 double-words are programmed without verifying the Flash location.

Programming/erase time

7

Short programming and erasing time & small page size => Advantage for data EEPROM emulation

Parameter	Typical value
64-bit programming time	82µ s
Page (4 Kbytes) erase time	22 ms
One row (64 double-word) programming time	Standard mode: 5.2 ms Fast mode: 3.8 ms
One page (4 Kbytes) programming time	Standard mode: 41.8 ms Fast mode: 30.4 ms
Flash (1 Mbyte) programming time	Standard mode: 11 s Fast mode: 8 s
Mass erase time	22 ms

life.augmented

The Flash memory programming time is only 82 µs for 64-bit double-words. To program one page (4 Kbytes), 41.8 ms are needed in Standard mode and 30.4 ms in Fast mode. For the complete Flash memory to be programmed, it requires 8 s in Fast mode.

The page erase time is 22 ms. It also requires only 22 ms to erase the complete Flash memory.

The short programming and erase time, plus the small page size, make it convenient for data EEPROM emulation.

Row (64 double-word) Fast programming

8

- Only the main memory can be programmed with Fast programming.
- Flash locations are not verified by HW before programming
- The 64 double-words must be written successively.
  - The high voltage is kept on the Flash memory for all programming.
  - Maximum time between two double-word write requests is the programming time (approx. 20 µs) => Interrupts should be disabled
- The Flash clock frequency (HCLKS) must be at least 8 MHz.

life.augmented

A fast programming mode allows to program 64 double-words faster than in standard programming mode.
Only the main memory can be programmed in Fast programming mode.
The Flash memory address location contents are not verified by hardware before programming in Fast mode.
The 64 double-words must be written successively. The high voltage is kept on the Flash memory for all programming. The maximum time between two double-word write requests is the programming time, which is approximately 20 µs. Consequently, interrupts should be disabled to ensure that the 20 µs between the two word write requests is not exceeded.
The minimum clock frequency must be at least 8 MHz in Fast programming mode.

Standard versus fast programming modes

9

	Programming mode	
	Standard	**Fast**
Target	Main memory + OTP area	Main memory only
Granularity	8 bytes	512 bytes
Specific limitations	None	No check of address location Flash clock frequency≥ 8 MHz Interrupts prohibited
Time to program 512 bytes	5.2 ms	3.8 ms

life.augmented

This slide compares Standard and Fast programming modes. Standard mode can be used to program the main memory and OTP areas while Fast mode cannot be used for OTP programming. Standard mode allows programming 64-bit double-words, or 8 bytes, whereas Fast mode only allows programming 64-bit double-words, or only 512 bytes.
In Fast mode, the address location content is not checked before programming, the Flash clock frequency must be greater than 8 MHz and CPU interrupts are prohibited. It takes 5.2 ms to program 512 bytes in Standard mode and 3.8 ms in Fast mode.

Flash memory retention

Endurance	10 Kcycles minimum @ -40 to +105 C
Data retention	30 years after 10 Kcycles at 55° C 15 years after 10 Kcycles at 85° C 10 years after 10 Kcycles at 105° C 30 years after 1 Kcycle at 85° C 15 years after 1 Kcycle at 105° C 7 years after 1 Kcycle at 125° C
Cycling retention	1 ppm

life.augmented

The Flash memory is guaranteed for a minimum of 10 000 cycles up to 105 °C. Data retention is 30 years after 10 000 cycles at 55 °C, 15 years after 10 000 cycles at 85 °C and 10 years after 10 000 cycles at 105 °C. It is 30 years after 1 000 cycles at 85 °C, 15 years after 1 000 cycles at 105 °C and 7 years after 1 000 cycles at 125 °C.

Flash memory read access

11

80 DMIPS at 64 MHz

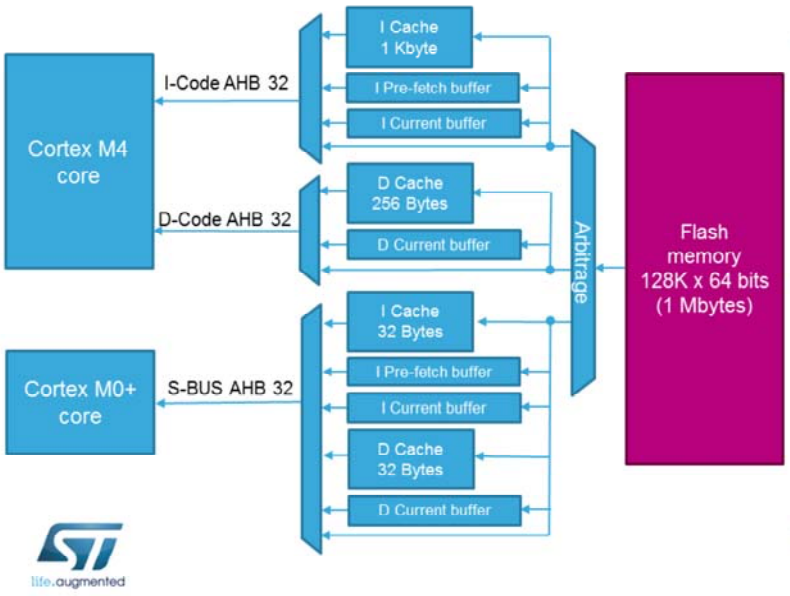
- Adaptive real-time memory accelerator (ART Accelerator™) allowing linear performance versus frequency, regardless of Flash memory access time.

Wait states (WS) (Latency)	HCLK (MHz)	
	V_{CORE} Range 1	V_{CORE} Range 2
0 WS	≤ 18	≤ 6
1 WS	≤ 36	≤ 12
2 WS	≤ 54	≤ 16
3 WS	≤ 64	

In order to read the Flash memory, it 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 voltage scaling range. In Range 1, the Flash memory can be accessed up to 64 MHz with 3 wait states. It can be accessed with 0 wait states up to 18 MHz. For Range 2, it is up to 16 MHz, with 2 wait states. Thanks to the adaptive real-time memory accelerator, the ART accelerator, the program can be executed with 0 wait states independent of the clock frequency. This provides an almost linear performance in relation to frequency and allows to reach 80 Dhrystone MIPS at 64 MHz.

Adaptive real-time memory accelerator (ART Accelerator™)

Outstanding performances and low-power



- Cortex-M4
  - **Instruction cache** = 32 lines of 4x64 bits (1 Kbyte), for instruction
  - **Data cache** = 8 lines of 4x64 bits (256 bytes), for literal pools
  - **Pre-fetch buffer**
- Cortex-M0+
  - **Instruction cache** = 4 lines of 1x64 bits (32 bytes), for instructions
  - **Data cache** = 4 lines of 1x64 bits (32 bytes), for literal pools
  - **Pre-fetch buffer**
- Best tradeoff between cache size, power and performance

The ART accelerator brings outstanding performance and reduces dynamic power consumption. It consists of a Cortex-M4 1-Kbyte instruction cache, 256 bytes of data cache and a pre-fetch buffer, and a Cortex-M0+ 32-bytes instruction cache, 32 bytes of data cache and a pre-fetch buffer.
The Cortex-M4 instruction cache contains 32 lines of 4 double-words and the data cache has 8 lines of 4 double-words. Once all the instruction cache memory lines have been filled, the LRU (least recently used) policy is used to determine the line to replace in the instruction memory cache. This feature is particularly useful when code contains loops.
This architecture is chosen to provide the best tradeoff between cache size, power consumption and performance.

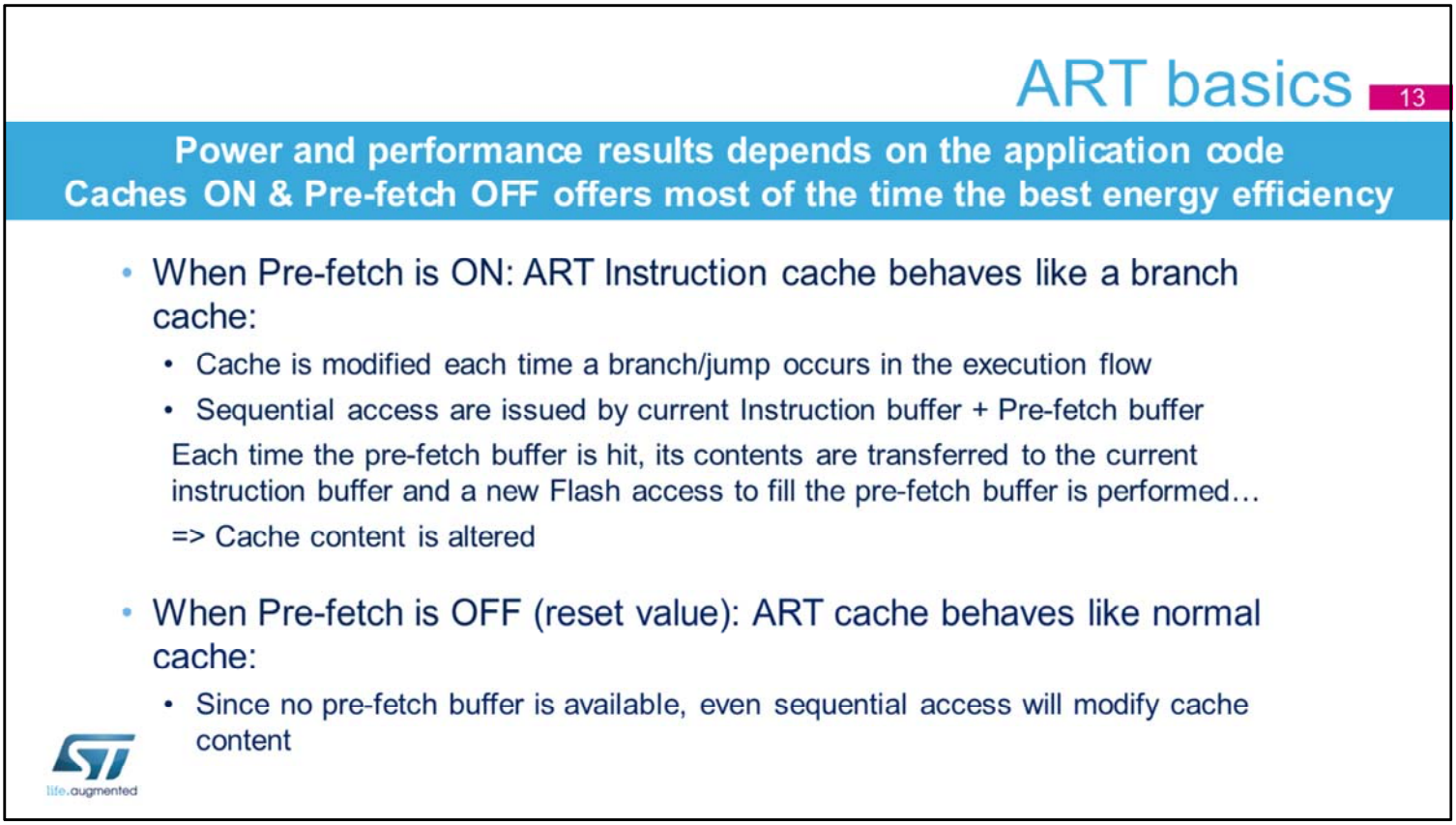
After each miss, the cache is updated with only the requested double-word in order to limit the Flash access for power-saving. In a line, the 4 double-words may not all be valid.

In case of a miss, the code takes the instruction directly from the Flash memory. In parallel, the 64-bit line is copied into the current buffer enabled and I-Cache if enabled. So the next sequential access is taken directly from the current buffer.
If Pre-fetch is enabled, another 64-bit Flash memory access is performed to fill the Pre-fetch buffer with sequential data.

When the data is present in the current buffer, the CPU reads the current buffer. The next sequential read is performed in the Pre-fetch buffer, which is copied into the current buffer, so that it is free to be filled with the next sequential data.

If the data is not present in the current buffer, it is read from the Pre-fetch buffer if it is present. If not, it is read from the instruction cache if there is a cache hit. Otherwise, a Flash access is performed.

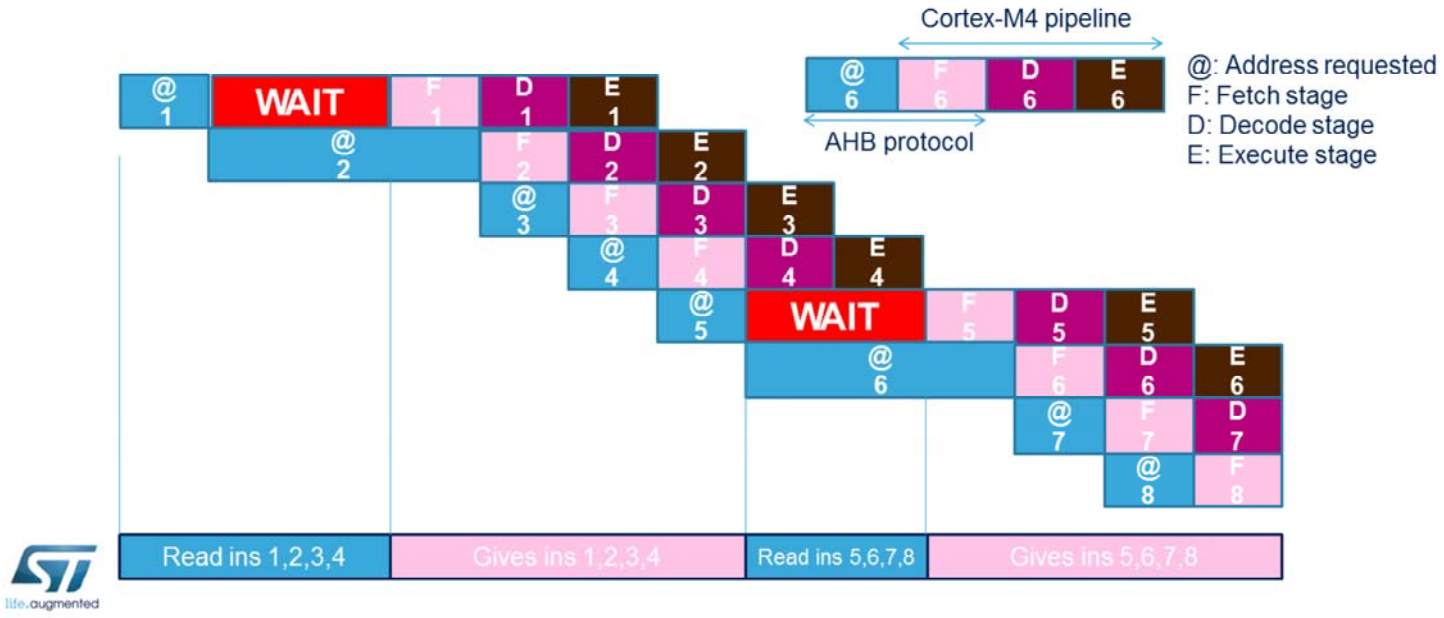
Flash access arbitration between Cortex-M4 I-Code instructions, D-Code data and Cortex-M0+ S-bus Instructions and data uses Round-robin.



The Instruction Cache behaves differently depending on if the pre-fetch buffer is enabled or not.
If the pre-fetch buffer is enabled, the ART instruction cache behaves like a branch cache. The cache is modified each time a branch or a jump occurs in the execution flow. Sequential accesses are issued by the current instruction buffer and the pre-fetch buffer; each time the pre-fetch buffer is hit, its contents are transferred to current instruction buffer and a new Flash access to fill the pre-fetch buffer is performed … In this case, the cache content is not altered.
If the pre-fetch buffer is disabled, the ART instruction cache behaves like a normal cache. Since no pre-fetch buffer is available, even a sequential access will modify the cache content.
The power and performance tradeoff must be evaluated for each application to know whether is it is better to enable or disable the pre-fetch buffer. For most of applications, enabling

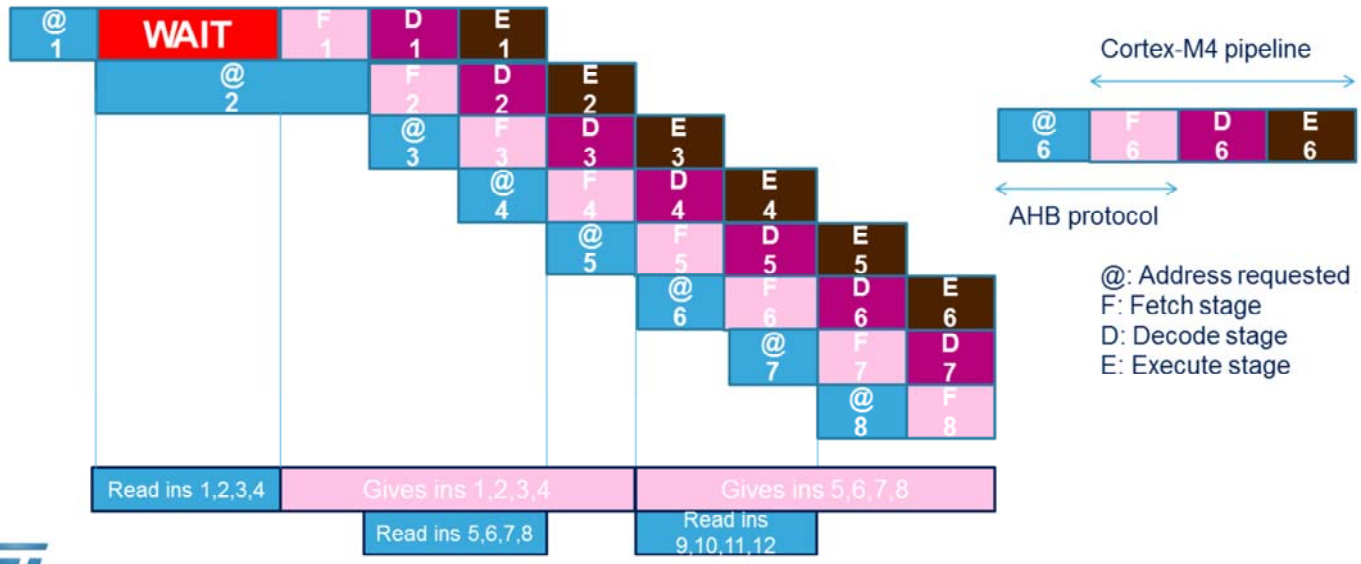
the pre-fetch buffer allows to increase slightly the performance but with a higher consumption. Most of the time, the best energy efficiency is provided with caches enabled and pre-fetch buffer disabled, as it often reduces the number of Flash memory accesses.

Sequential 16-bit instruction execution (2 WS), without pre-fetch

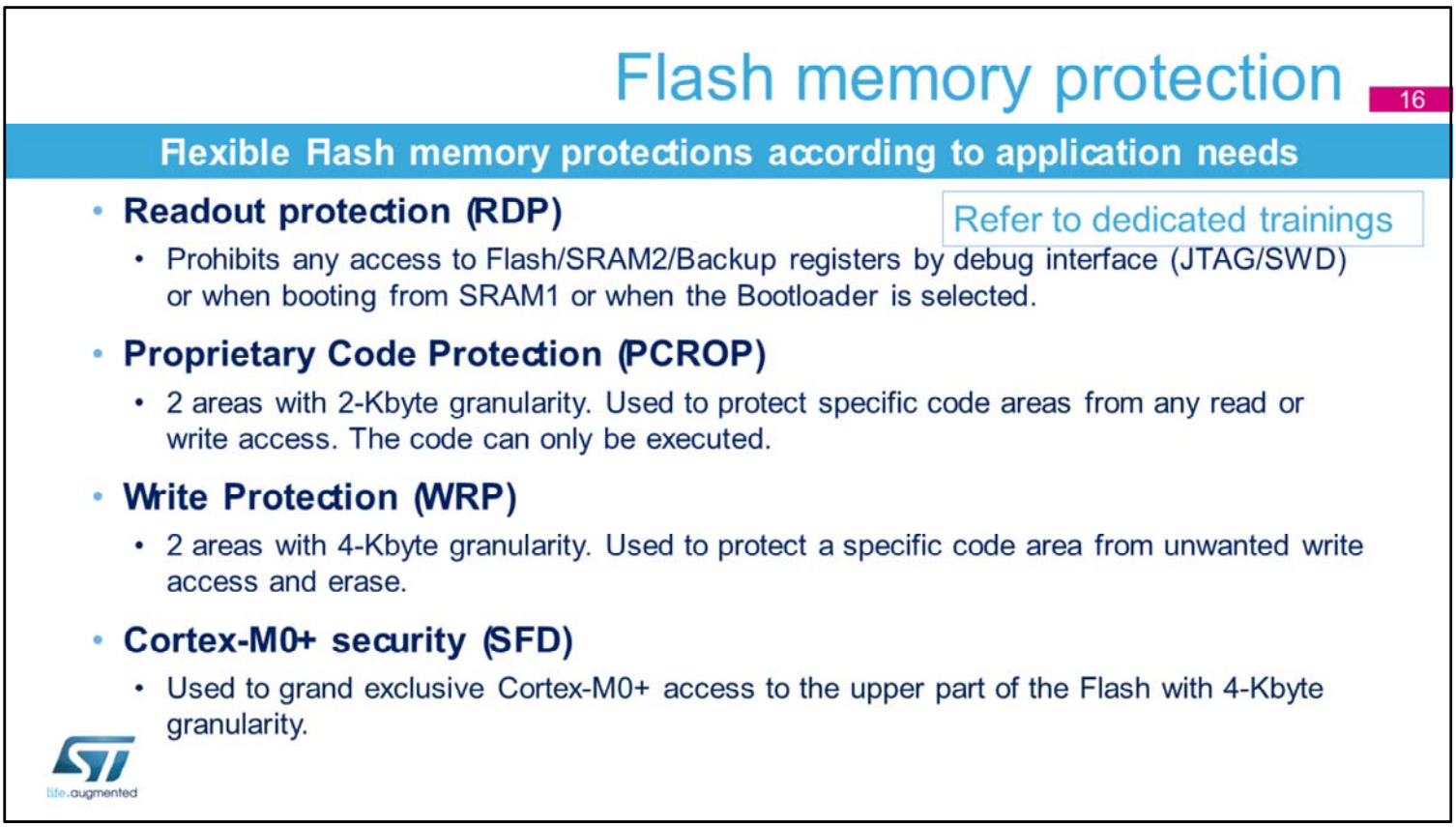


This slide shows the number of cycles needed to execute sequential 16-bit instructions without pre-fetch, when 2 wait states are needed to access the Flash memory. Every Flash access provides 64 bits or 4 instructions; 2 wait states are therefore inserted every 4 instructions, at every Flash access.

Sequential 16-bit instruction execution (2 WS), with pre-fetch



This slide shows the number of cycles needed to execute sequential 16-bit instructions with pre-fetch enabled, when 2 wait states are needed to access the Flash memory. After each Flash access, another Flash access is performed to fill the pre-fetch buffer. So after all instructions are fetched from the current buffer, the next sequential instruction is read from the pre-fetch buffer and no wait state is inserted as long as the instruction flow is sequential.

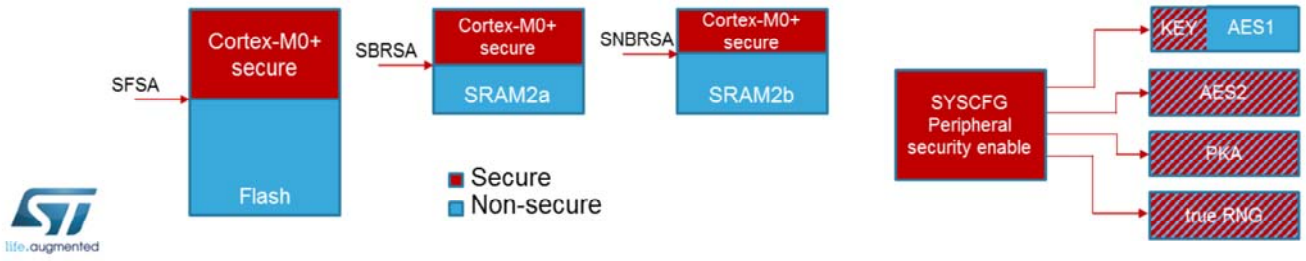


Several Flash memory protection options can be configured using the option bytes.
The readout protection is configured using the RDP option byte. The readout protection prohibits any access to the Flash memory, the SRAM2 and the backup registers by the debug interface or when booting from SRAM1 or when the bootloader is selected.
The proprietary code protection is configured using the PCROP option bytes. These options protect specific code areas from any read or write access; the code can only be executed. The protected areas can be defined with 2-Kbyte granularity and two areas can be defined.
The write protection is configured using the WRP option bytes. These options protect specific code areas from unwanted write access and erase. The write-protected area can be defined with 4-Kbyte granularity.
The Cortex-M0+ security is configured using the SFD option byte. This option secures a specific Flash memory area for exclusive Cortex-M0+ access. The Cortex-M0+ security area can be defined

with a 4-Kbyte granularity.
Please refer to the specific trainings about System protections and Cortex-M0+ security for more details about these protection options.

Cortex-M0+ security

- The upperpart of the Flash memory can be secured for exclusive Cortex-M0+ access.
  - Defined by secure user options SFD and SFSA.
- Global security enabled.
  - Allows to add SRAM2a upperpart security via secure user option SBRD and SBRSA
  - Allows to add SRAM2b upperpart security via secure user option SNBRD and SNBRSA
  - Allows peripheral security enabled in SYSCFG.



Cortex-M0+ security is enabled by clearing the Flash Security Disable option. The secure Flash starts from the address in the Secure Flash Start Address option. In addition to the Flash memory, the SRAM2a and SRAM2b security can also be enabled by Secure Backup RAM Disable, Secure Backup RAM Start Address, Secure Non Backup RAM Disable, and Secure Non Backup RAM Start Address options. Security peripherals like Advanced Encryption Standard accelerator, Private Key Accelerator, and True Random Number Generator may be secured by register bits in the system Configuration IP.

User option bytes

The user option bytes are loaded:

- after a Power reset (BOR or exit from Standby/Shutdown)
- when the OBL_LAUNCH bit is set in the Flash control register (FLASH_CR).

Options	Description
BOR_LEV[2:0]	Brown-out reset threshold level
nRST_STOP; nRST_STDBY; nRST_SHDW	Reset No Reset generated when exiting Stop Standby Shutdown mode
WWDG_SW; DWG_SW IWDG_STOP; IWDG_STDBY	Hardware Software window watchdog / independent watchdog Independent watchdog counter is frozen / not frozen in Stop Standby mode
nBOOT0, nBOOT1 BOOT0SW	Boot configuration BOOT0 selection
SRAM_RST SRAM_PE	SRAM2a2b Erase when system reset SRAM2a2b parity check enable
PCCDBA	IPCC data buffer base address

life.augmented

Several user option bytes are available in the Flash memory to configure certain specific features of the device.
The user option bytes are loaded in two cases: either after a power or brown-out reset, when exiting from Standby or Shutdown modes, or when the OBL_LAUNCH bit is set in the Flash control register.
Three option bits are used to configure the brown-out reset threshold.
Three options are available to prohibit or allow the Stop, Standby and Shutdown low-power modes.
Four options configure if 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.
Three options are used together with the BOOT0 pin to configure the memory used for booting.
Two options are used to configure if the SRAM2 is erased with

the system reset, and to enable the SRAM2 parity check. One option is used to define the common memory area in SRAM2 for Inter Processor Communication data buffers.

User option bytes

Options	Description
RDP[7:0]	Readout protection level
PCROP1A_STRT PCROP1A_END PCROP1B_STRT PCROP1B_END	PCROP area A start offset address PCROP area A end offset address PCROP area B start offset address PCROP area B end offset address
PCROP_RDP	PCROP area preserved when RDP level decreased
WRP1A_STR WRP1A_END WRP1B_STRT WRP1B_END	Write protection area A start offset address Write protection area A end offset address Write protection area B start offset address Write protection area B end offset address
SFSA, SBRSA, SNBRSA SBRV,C2BOPT	Cortex-M0+ Flash and SRAM2 security start addresses Cortex-M0+ boot reset vector and boot option Flash&SARM2.
DDS	Cortex-M0+ debug disable

* Option is Cortex-M0+ write secure

Several option bytes are used for memory protection options: the RDP for readout protection, PCROP for the start and end addresses of two areas, and WRP for the start and end addresses for each of the two areas.
The PCROP_RDP bit is used to preserve or erase the PCROP area when the readout protection is removed from Level 1 to Level 0.
The Cortex-M0+ secure memory areas in Flash memory is defined by the SFSA, and in SRAM2a by the SBRSA and SRAM2b by SNBRSA. The Cortex-M0+ reset vector is defined by SBRV and C2BOPT.
Debugging of the Cortex-M0+ is disabled by DDS.
The Cortex-M0+ secure memory area, boot options and debug disable have exclusive Cortex-M0+ write access. They can be read by the Cortex-M4 to provide information on the secure memory areas.

Interrupts

Interrupt event	Description
Interrupts	
End of operation	Set by hardware when one or more Flash memory operations (programming / erase) is completed successfully.
Operation error	Set by hardware when a Flash memory operation (program / erase) is unsuccessful.
Read error	Set by hardware when an address to be read through the D-bus belongs to a Read-protected area of the Flash memory (PCROP protection).
ECC correction	Set by hardware when one ECC error has been detected and corrected.
Non-maskable interrupt (NMI)	
ECC detection	Set by hardware when two ECC errors have been detected.

life.augmented

Four interrupts can be generated by the Flash memory.
The end-of-operation interrupt, which is triggered when one or more Flash program or erase operations is completed successfully.
The operation error interrupt is triggered when a Flash memory program or erase operation failed.
The read error interrupt is triggered when an address read through the Core Data Bus belongs to an area of the Flash protected by the PCROP option.
The ECC interrupt is triggered when one ECC error is detected and corrected. When two ECC errors are detected, a non-maskable interrupt is generated.

Consumption optimization when execution from SRAM

- Flash clock can be gated off in Run/Low-power run and/or in Sleep/Low-power sleep modes
  - Flash clock is configured in the Reset and Clock Controller (RCC)
  - Flash clock is enabled by default
- Flash memory can be configured in Power-down mode during Sleep/Low-power sleep modes
- Flash memory can be configured in Power-down mode during Run/Low-power run modes

life.augmented

The Flash memory's consumption can be reduced when the code is not executed from Flash memory.
The Flash clock can be gated off in Run and Low-power run modes. It can also be configured to be gated off in Sleep and low-power sleep modes. The Flash clock is configured in the Reset and Clock controller. It is enabled by default.
The Flash memory can be configured in Power-down mode during the Sleep and low-power sleep modes.
It can also be configured in Power-down mode during Run and Low-power run modes, when the code is executed from SRAM.
Gating the clock and putting the Flash memory in Power-down mode significantly reduces power consumption.

Low-power modes

Mode	Description
Run	Active. Flash clock can be disabled if code is executed fom SRAM and the Flash memory is in Power-down mode.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode. Flash clock can be disabled during Sleep mode. Flash memory can be put in Power-down mode.
Low-power run	Active. Flash clock can be disabled if code is executed fom SRAM and the Flash memory is in Power-down mode.
Low-power sleep	Active. Peripheral interrupts cause the device to exit Low-power sleep mode. Flash clock can be disabled during Low-power sleep mode. Flash memory can be put in Power-down mode.
Stop 0/Stop 1/Stop 2	Flash clock of. Contents of peripheral registers are kept.
Standby	Powered-down. The Flash memory interface must be reinitialized afer exiting Standby mode.
Shutdown	Powered-down. The Flash memory interface must be reinitialized afer exiting Shutdown mode.

life.augmented

In Run and Low-power run modes, the Flash memory is active. Its clock can be disabled if code is executed from SRAM and the Flash memory is in Power-down mode.
In Sleep and Low-power sleep modes, the Flash clock can be disabled and the Flash memory configured in Power-down mode.
In Stop 0, Stop 1 and Stop 2 modes, the Flash clock is off. The content of the Flash interface registers is retained.
In Standby and Shutdown modes, the content of the Flash interface registers is lost and must be reinitialized after exiting the mode.

Flash memory performance

tbd Coremark / MHz

- Flash memory performance is almost linear with frequency thanks to ART accelerator: 3.32 CoreMark / MHz (Caches ON, Pre-fetch OFF) => 212.5 CoreMark at 64 MHz

		ART accelerator ON (Caches On, Pre-fetch Off)
Range 1 @ 64 MHz (2 wait states)	Consumption (µA/MHz) (SMPS OFF)	125
	Performance (CoreMark/MHz)	3.32
	Energy efficiency (CoreMark/mA)	26.6
Range 2 @ 16 MHz (1 wait state)	Consumption (µA/MHz)	116
	Performance (CoreMark/MHz)	2.48
	Energy efficiency (CoreMark/mA)	21.4

life.augmented

The performance of the Flash memory is almost linear with the frequency using the ART accelerator. The CoreMark score is 212.5 at 64 MHz, which corresponds to 3.32 CoreMark / MHz with the Instruction Cache, Data Cache enabled and Pre-fetch buffer is disabled.
In Range 2 at 16 MHz, the performance is 2.48 CoreMark / MHz with the Instruction and Data caches enabled and the Pre-fetch buffer disabled. In Energy efficiency at 64 MHz with Switched Mode Power Supply off is 26.6 CoreMark/mA, and in Range 2 energy efficiency lowers only to 21.4 CoreMark/mA at 16 MHz.

Flash memory sharing

- The Flash memory is shared between both the Cortex-M4 and the Cortex-M0+.
  - Access arbitration is handled in the ART according to Round Robin.
    - Read and fetch access take priority over write and erase.
    - Read access take priority over fetch from the other core.
- Shared performance.
  - Low impact due to instruction and data caches for both Cortex-M4 and Cortex-M0+.

CortexM4	CortexM0+	ART accelerator ON (Caches On, Pre-fetch Off)		Impact vs stand-alone
@ 64 MHz	@ 32 MHz	Cortex-M4 Performance (CoreMark/MHz)	3.28	-1.30%
		Cortex-M0+ Performance (CoreMark/MHz)	1.68	-0.15%
@ 16 MHz	@ 16 MHz	Cortex-M4 Performance (CoreMark/MHz)	2.48	-0.04%
		Cortex-M0+ Performance (CoreMark/MHz)	1.48	-0.01%

life.augmented

The Flash memory is shared between the Cortex-M4 and the Cortex-M0+. Both CPUs use the Flash memory to execute instructions. The performance of the Flash memory has minimal impact due to the ART accelerator. During simultaneous code execution, the Cortex-M4 CoreMark/MHz is 3.28 at 64 MHz with Cortex-M0+ at 32 MHz, the Instruction Cache, Data Cache enabled and Pre-fetch buffer is disabled. This translates into a Cortex-M4 performance loss of only 1.3% The Cortex-M4 CoreMark/MHz score is 2.48 at 16 MHz with Cortex-M0+ at 16 MHz. This translates into a Cortex-M4 performance loss of only 0.04%.

Flash memory operations

- Program and erase operations are only possible in power Range 1.
- Since there is only a single bank, Flash memory program and erase operation blocks execution from Flash memory for both CPUs.
- Starting new Flash memory operations can be suspended with Program Erase Suspend (PESD)
  - Any ongoing operation will completed
  - Any new operation will be suspended until PESD bits are cleared.
  - Each CPU has its own PESD bit.

Flash memory program and erase operations are only possible in Power Range 1. In Range 2 and Low-power modes, Flash memory program and erase operations are prohibited.
Due to the single-bank Flash memory architecture, program and erase operations will block execution for both CPUs. To prevent Flash memory operation from impacting real-time CPU performance, they can be suspended. As long as the Suspend is active, no new operations will be started, guaranteeing the execution can continue. If an on-going Flash operation has been enabled before the Suspend, it will be completed. Each CPU can request a Flash operation Suspend using its own Suspend register bit.

Related peripherals

- Refer to these peripheral trainings linked to this peripheral
  - System configuration controller (SYSCFG)
  - Reset and clock controller (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC and AIEC)
  - System protections
  - Cortex-M0+ security

life.augmented

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

References

- For more details, please refer to the following document
  - AN2606: STM32 microcontroller system memory boot mode – Application note

life.augmented

For more details, please refer to application note AN2606 about the STM32 microcontroller system memory boot mode.