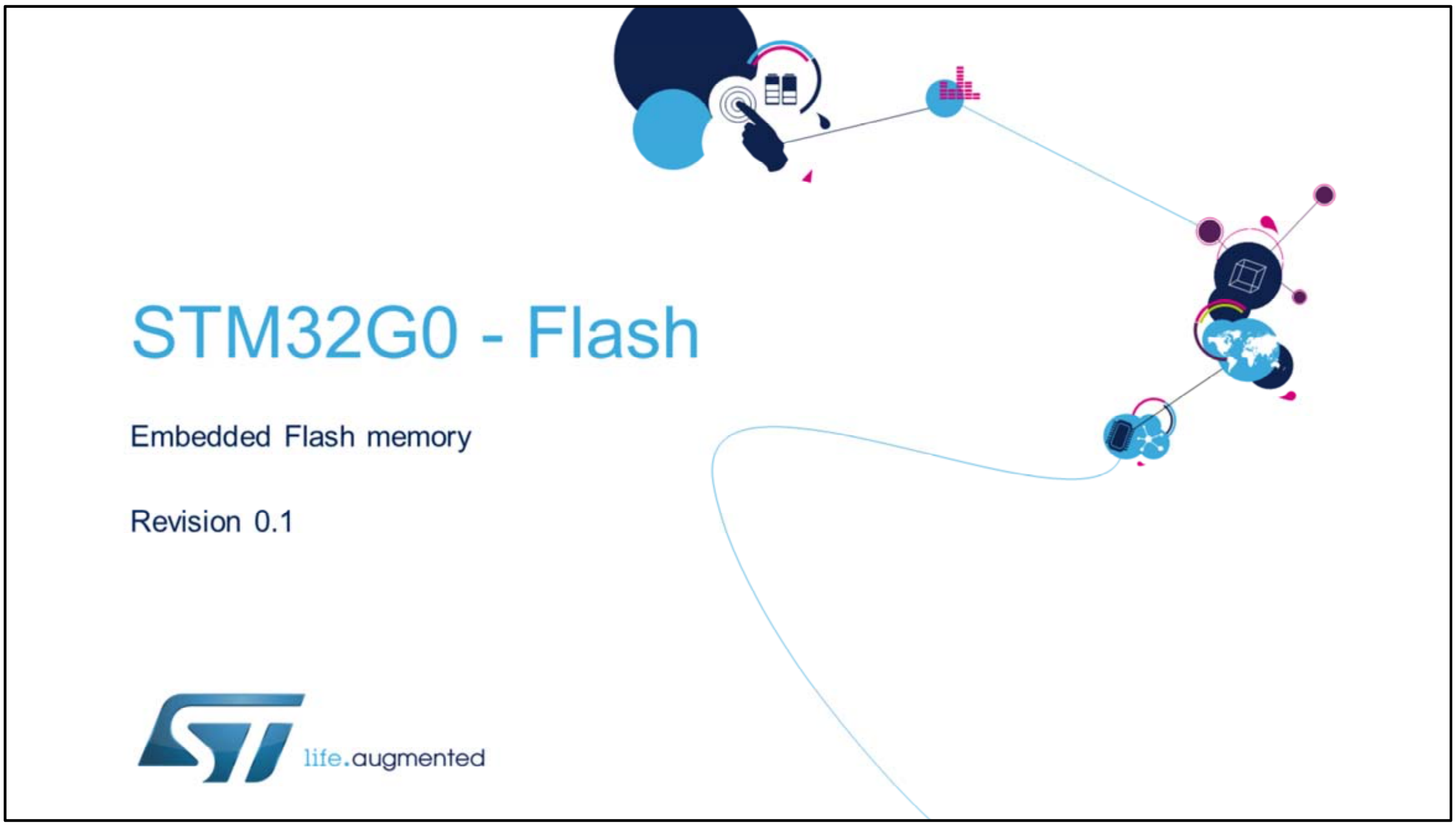


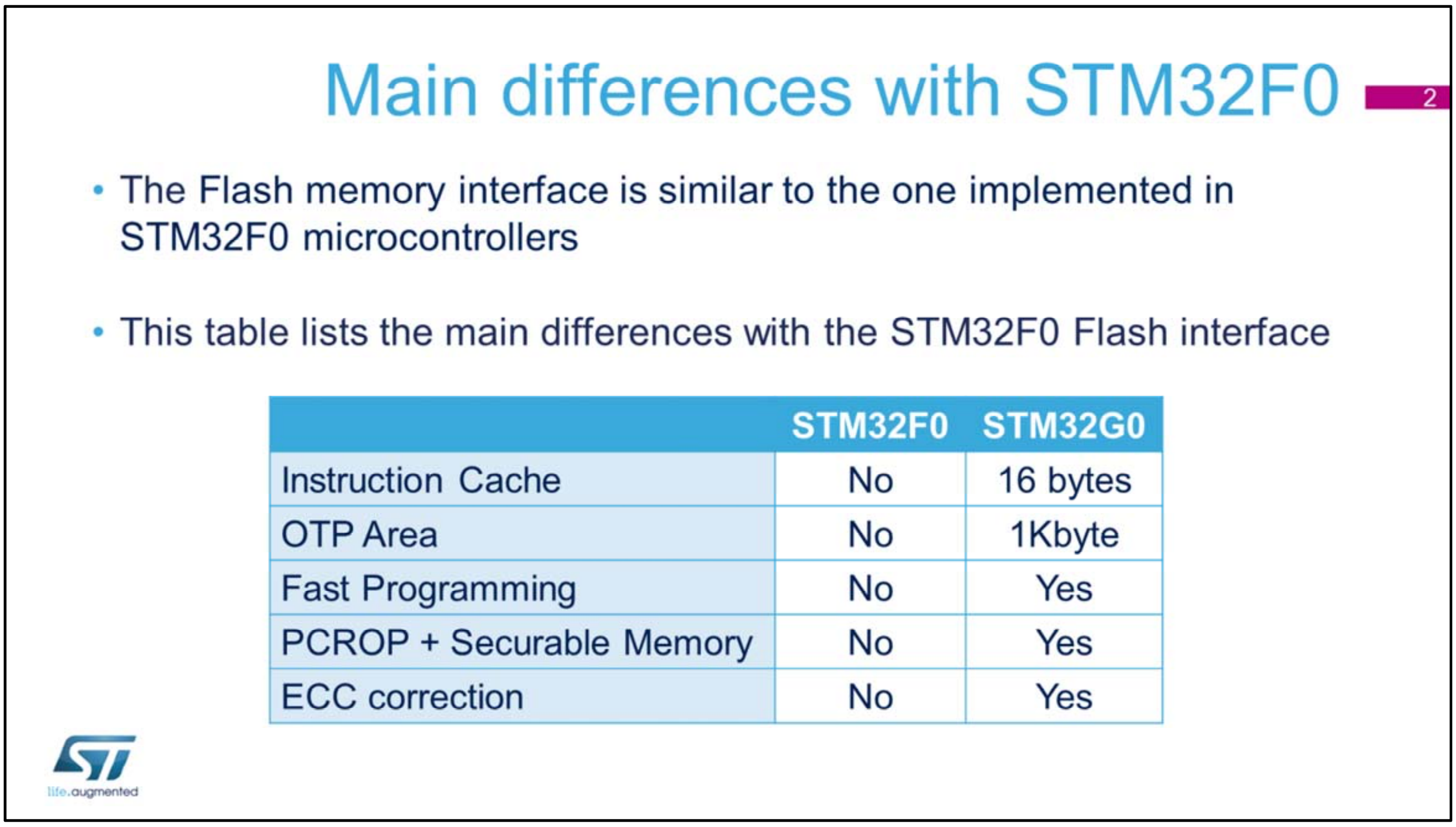
STM32G0 - Flash

Embedded Flash memory

Revision 0.1

life.augmented

Hello, and welcome to this presentation of the embedded Flash memory which is included in all products of the STM32G0 microcontroller family.



Main differences with STM32F0

2

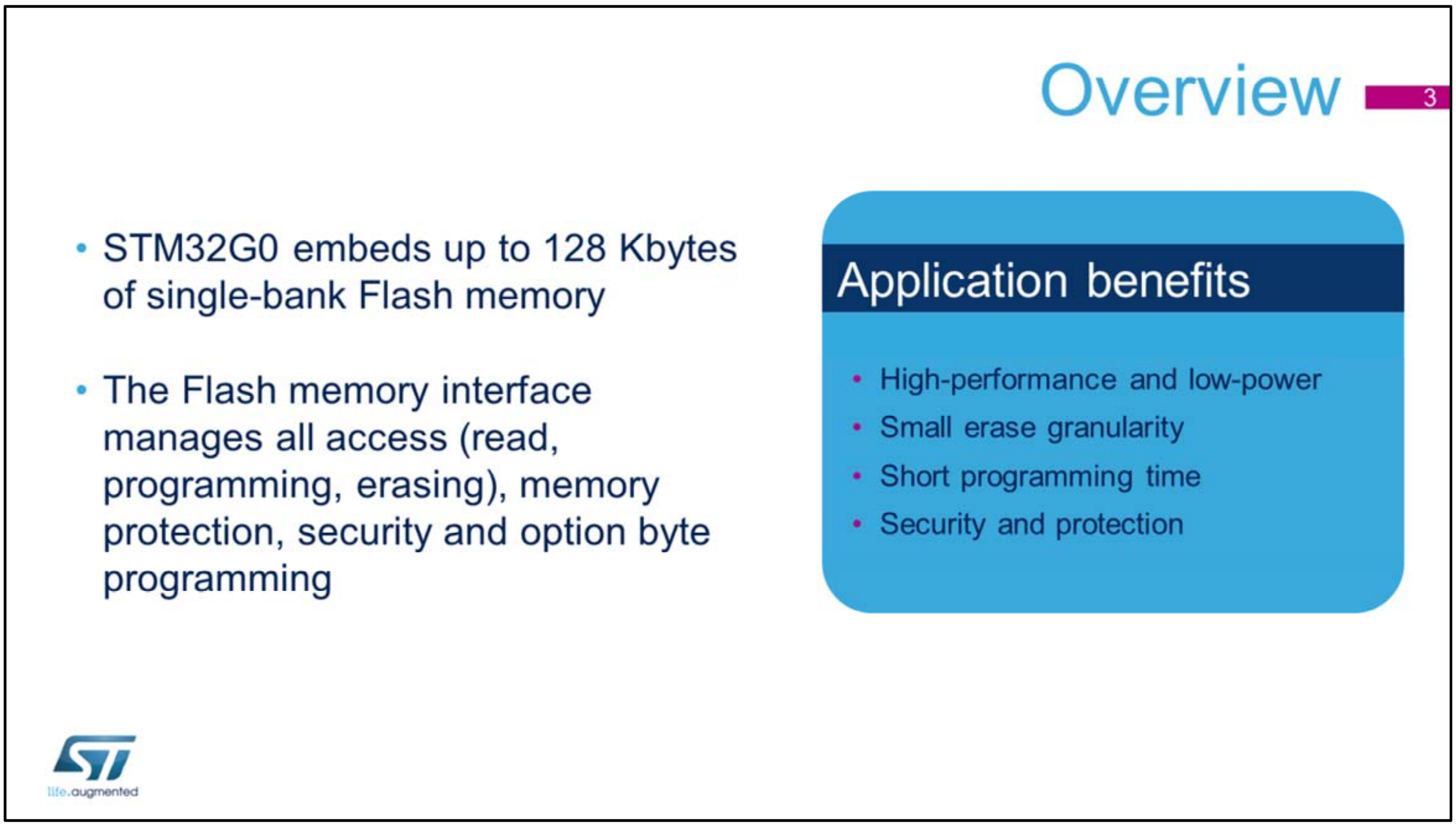
- The Flash memory interface is similar to the one implemented in STM32F0 microcontrollers
- This table lists the main differences with the STM32F0 Flash interface

	STM32F0	STM32G0
Instruction Cache	No	16 bytes
OTP Area	No	1Kbyte
Fast Programming	No	Yes
PCROP + Securable Memory	No	Yes
ECC correction	No	Yes

life.augmented

The STM32G0's Flash memory interface supports new features with regard to the STM32F0, as indicated in this table.
The cache and prefetch buffer decrease latency and consumption.
The One-time programming (OTP) area is used to store non-erasable data.
Fast Programming programs a row of 256 bytes instead of discrete 8-byte double words.
PCROP stands for proprietary code readout protection, which protects the code by only allowing the execution from Flash memory, but not reading or writing.
Securable memory cannot be called from non-secure areas. It is typically used to perform a secure boot with image authentication.
Error Correction and Checking (ECC) improves the reliability by detecting and eventually correcting bit flips

that may have occurred in the Flash memory. It is handled transparently by the Flash memory controller.



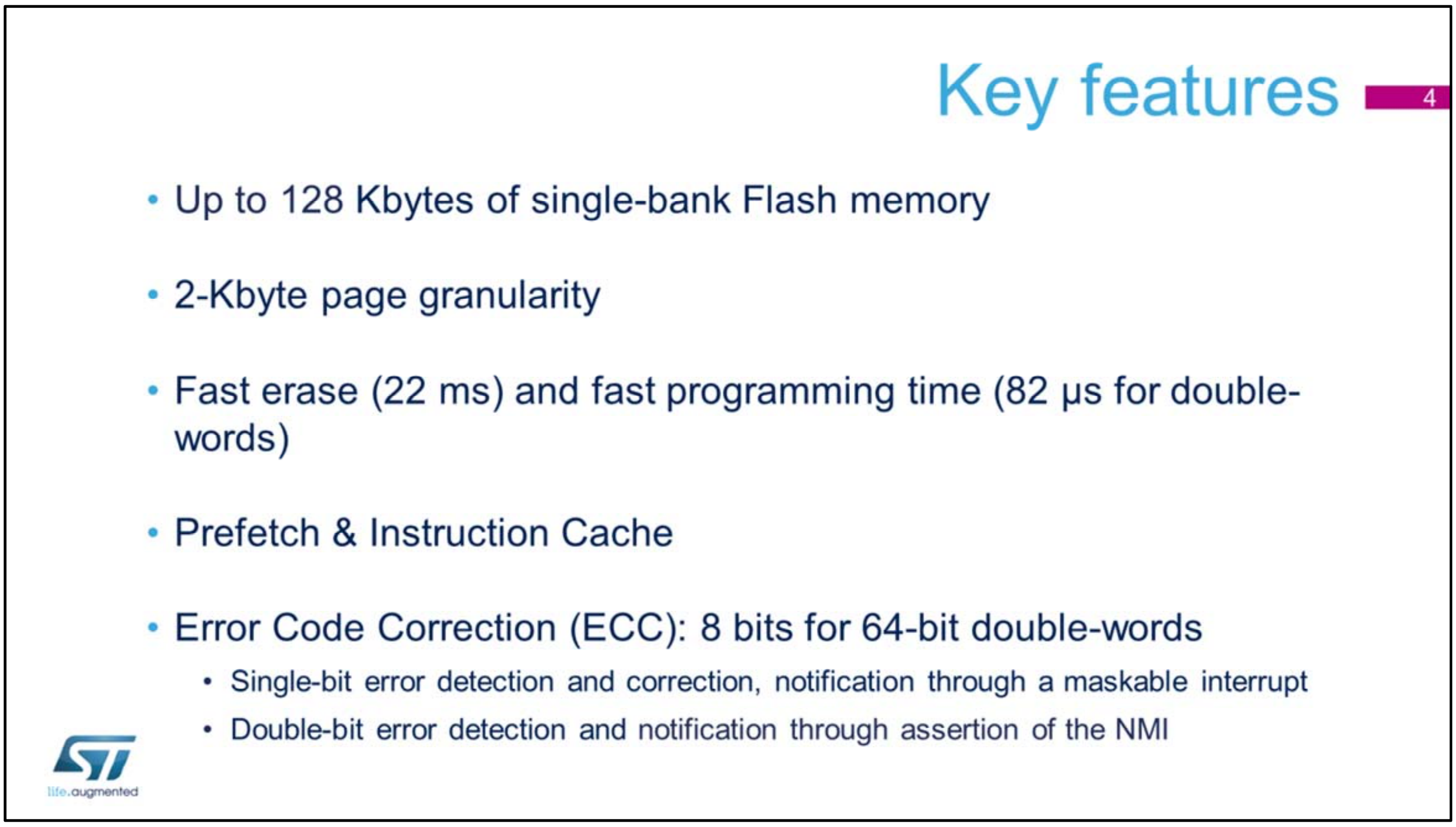
Overview

- STM32G0 embeds up to 128 Kbytes of single-bank Flash memory
- The Flash memory interface manages all access (read, programming, erasing), memory protection, security and option byte programming

Application benefits

- High-performance and low-power
- Small erase granularity
- Short programming time
- Security and protection

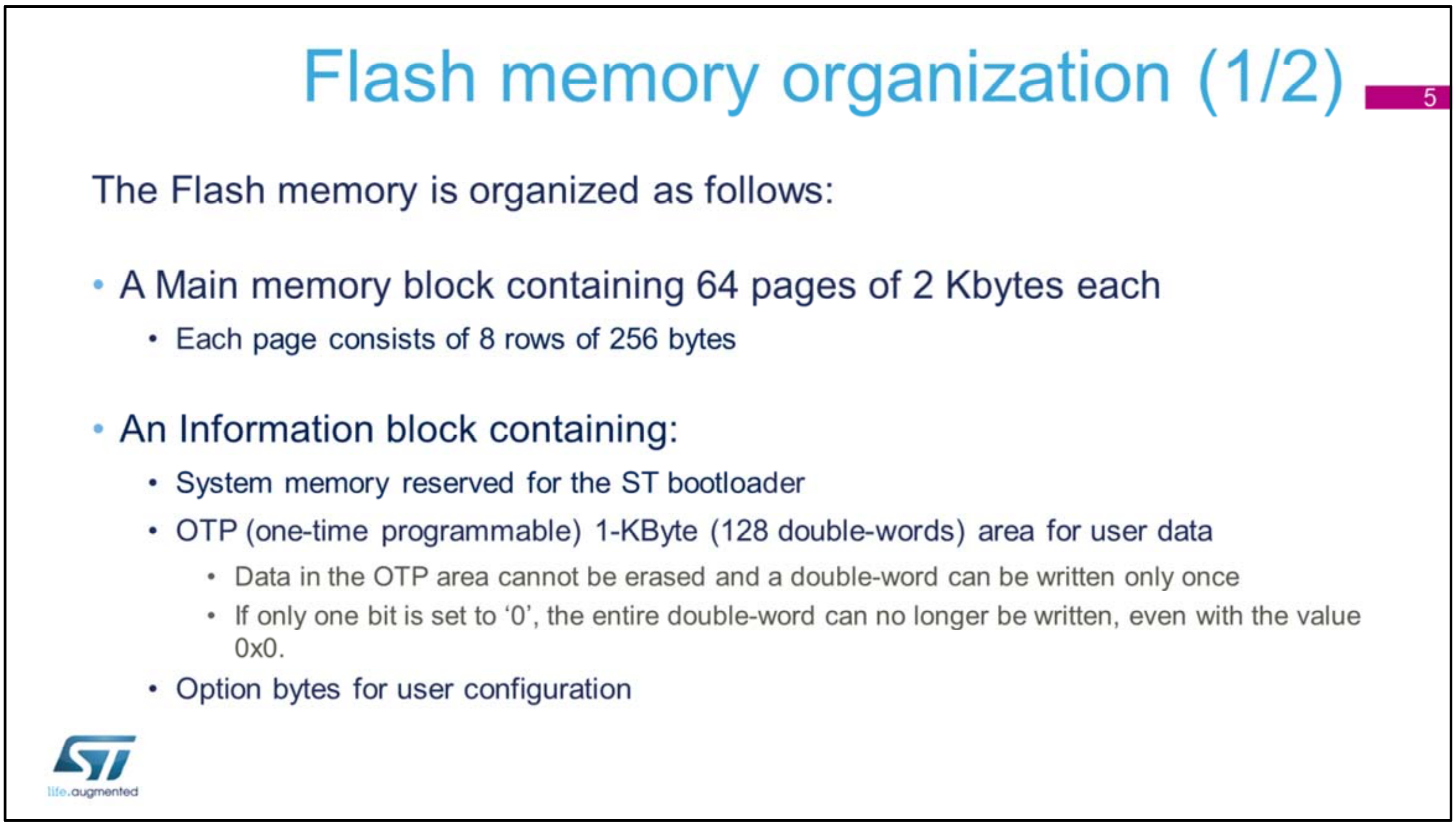
The STM32G0 embeds up to 128 Kbytes of single-bank Flash memory.
The Flash memory interface manages all memory access (read, programming and erasing) as well as memory protection, security and option bytes.
Applications using this Flash memory interface benefit from its high performance together with low-power access. It has a small erase granularity and short programming time.
It provides various security and protection mechanisms for code and data, read and write access.



Key features

- Up to 128 Kbytes of single-bank Flash memory
- 2-Kbyte page granularity
- Fast erase (22 ms) and fast programming time (82 µs for double-words)
- Prefetch & Instruction Cache
- Error Code Correction (ECC): 8 bits for 64-bit double-words
  - Single-bit error detection and correction, notification through a maskable interrupt
  - Double-bit error detection and notification through assertion of the NMI

The main Flash memory is split into 2-Kbyte pages that can be independently erased. A mass erase feature is also supported.
Flash memory access may require wait states according to the actual CPU frequency.
To reduce the latency, the Flash controller embeds both an 8-byte prefetch buffer and 16-byte instruction cache.
It also contributes to decrease the consumption, because they belong to the Vcore power domain.
An 8-bit ECC code is appended to the double-word to program. It is checked on read to detect and correct single-bit errors and detect double-bit errors.
In case of an uncorrectable error, the Flash memory controller asserts the Non-Maskable Interrupt (NMI) to the Cortex®-M0+.



Flash memory organization (1/2)

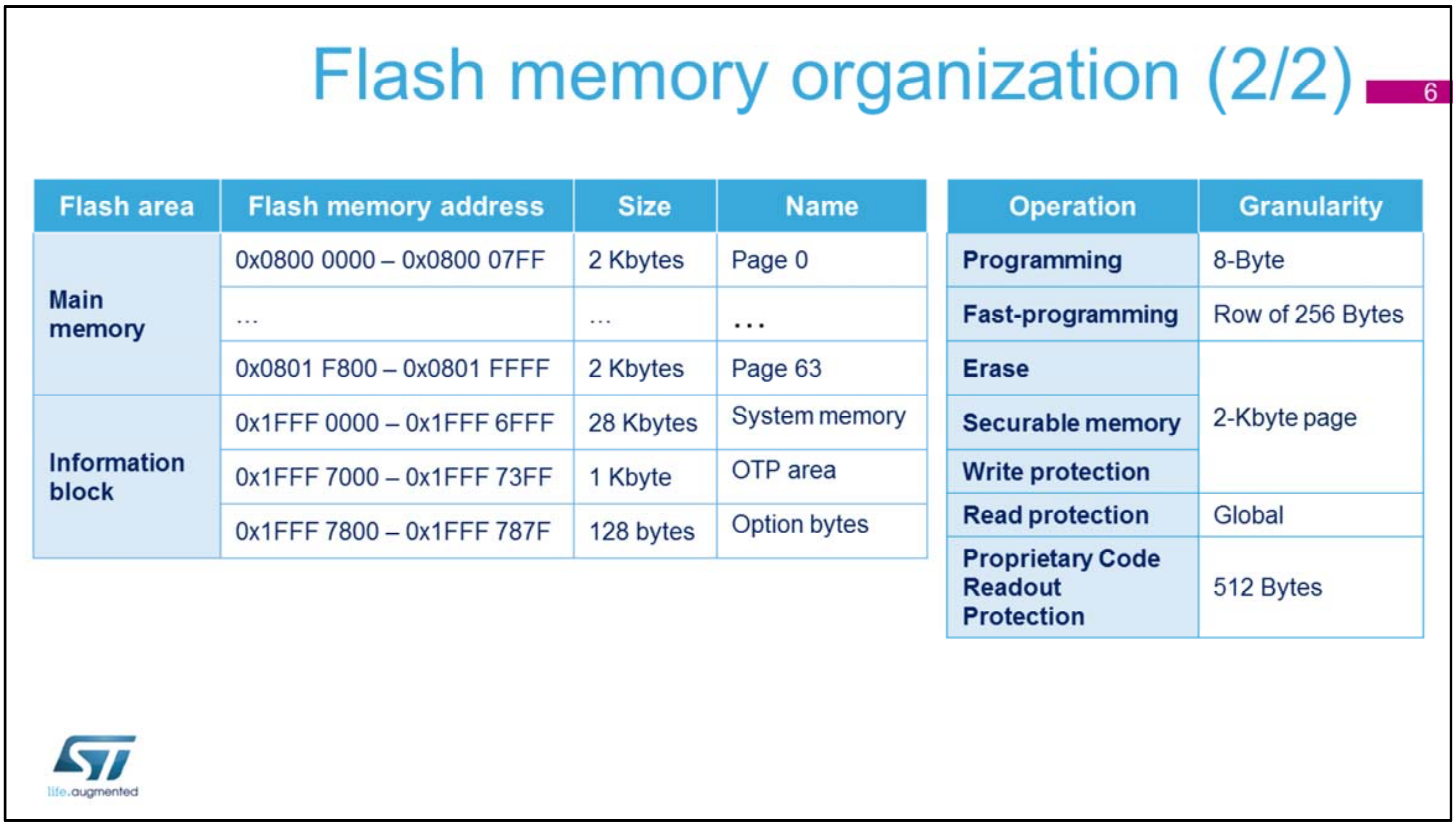
The Flash memory is organized as follows:

- A Main memory block containing 64 pages of 2 Kbytes each
  - Each page consists of 8 rows of 256 bytes
- An Information block containing:
  - System memory reserved for the ST bootloader
  - OTP (one-time programmable) 1-KByte (128 double-words) area for user data
    - Data in the OTP area cannot 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, even with the value 0x0.
  - Option bytes for user configuration

life.augmented

In addition to the 128 Kbytes of the main Flash memory, the STM32G0 supports:

- A System memory of 28 Kbytes containing the ST bootloader
- An OTP memory that can be used to store user data that cannot be erased
- Options bytes containing default settings to configure IPs in the system-on-chip. They are automatically loaded after a power-up reset.



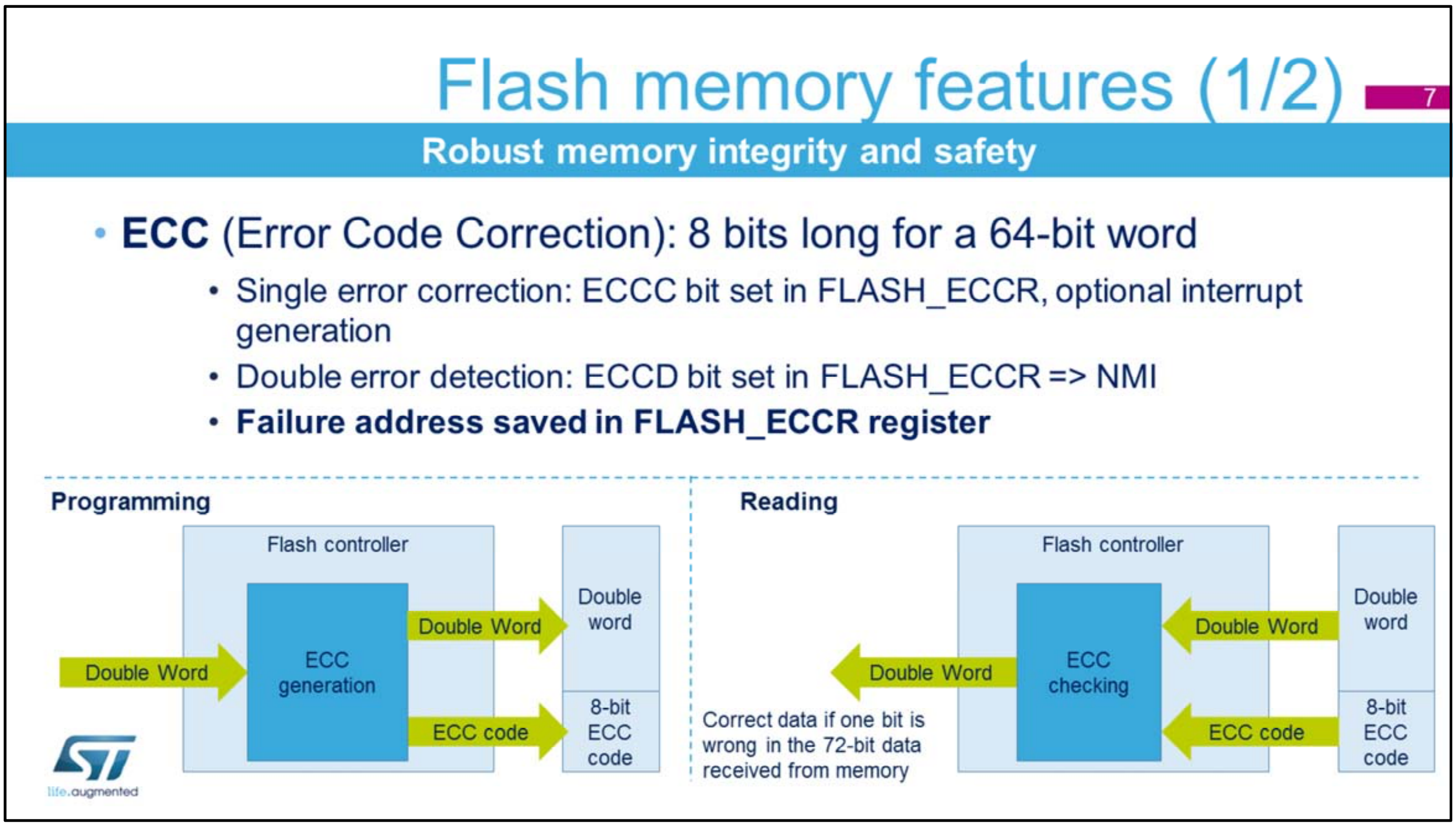
Flash memory organization (2/2)

Flash area	Flash memory address	Size	Name
Main memory	0x0800 0000 – 0x0800 07FF	2 Kbytes	Page 0
	…	…	…
	0x0801 F800 – 0x0801 FFFF	2 Kbytes	Page 63
Information block	0x1FFF 0000 – 0x1FFF 6FFF	28 Kbytes	System memory
	0x1FFF 7000 – 0x1FFF 73FF	1 Kbyte	OTP area
	0x1FFF 7800 – 0x1FFF 787F	128 bytes	Option bytes

Operation	Granularity
Programming	8-Byte
Fast-programming	Row of 256 Bytes
Erase	2-Kbyte page
Securable memory	
Write protection	
Read protection	Global
Proprietary Code Readout Protection	512 Bytes

The first table details the memory organization based on a Main Flash memory area and an information block.
The second table details the granularity of the Flash memory operations:

- Programming is done on 8-byte double words
- Fast programming is done on a row of 256 bytes
- Erase is done either globally (mass erase) or on 2-Kbyte pages
- The securable memory is aligned on pages.
- Write protection is done per page
- Read protection is global
- Proprietary Code Readout Protection is done on 512-byte areas



Flash memory features (1/2)

7

Robust memory integrity and safety

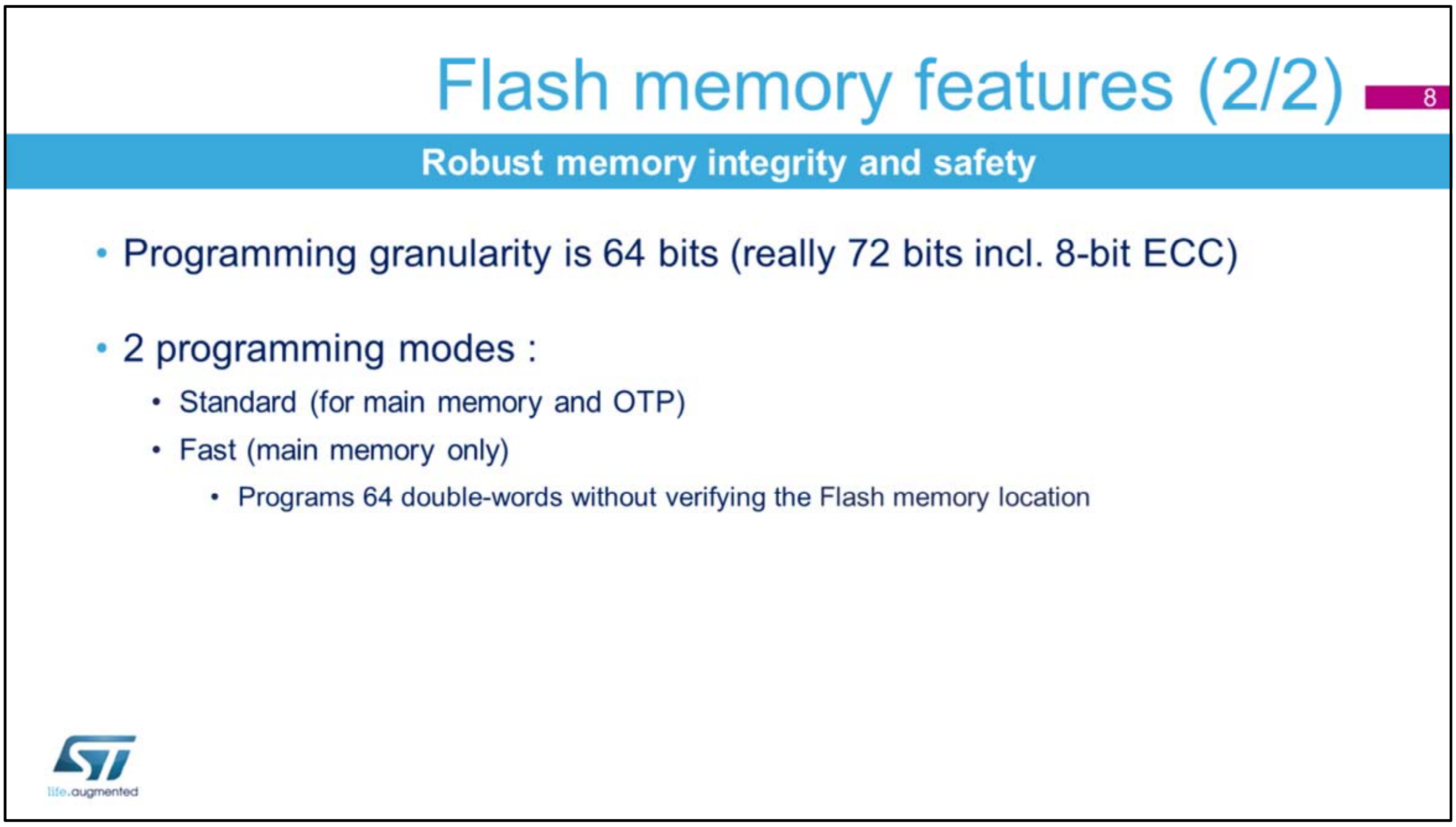
- **ECC** (Error Code Correction): 8 bits long for a 64-bit word
  - Single error correction: ECCC bit set in FLASH_ECCR, optional interrupt generation
  - Double error detection: ECCD bit set in FLASH_ECCR => NMI
  - **Failure address saved in FLASH_ECCR register**

Programming
Flash controller
Double Word
ECC generation
Double Word
ECC code
Double word
8-bit ECC code
Reading
Flash controller
Double word
8-bit ECC code
Double Word
ECC code
ECC checking
Double Word
Correct data if one bit is wrong in the 72-bit data received from memory
life.augmented

Data in Flash memory words are 72-bits wide: eight bits are added per each double word (64 bits). The ECC mechanism supports:

- One error detection and correction
- Two errors detection

When one error is detected and corrected, the ECCC flag (ECC correction) is set in the Flash ECC register (FLASH_ECCR). An interrupt can be generated.
When two errors are detected, the ECCD flag (ECC detection) is set in the Flash ECC register (FLASH_ECCR). In this case, an NMI is generated



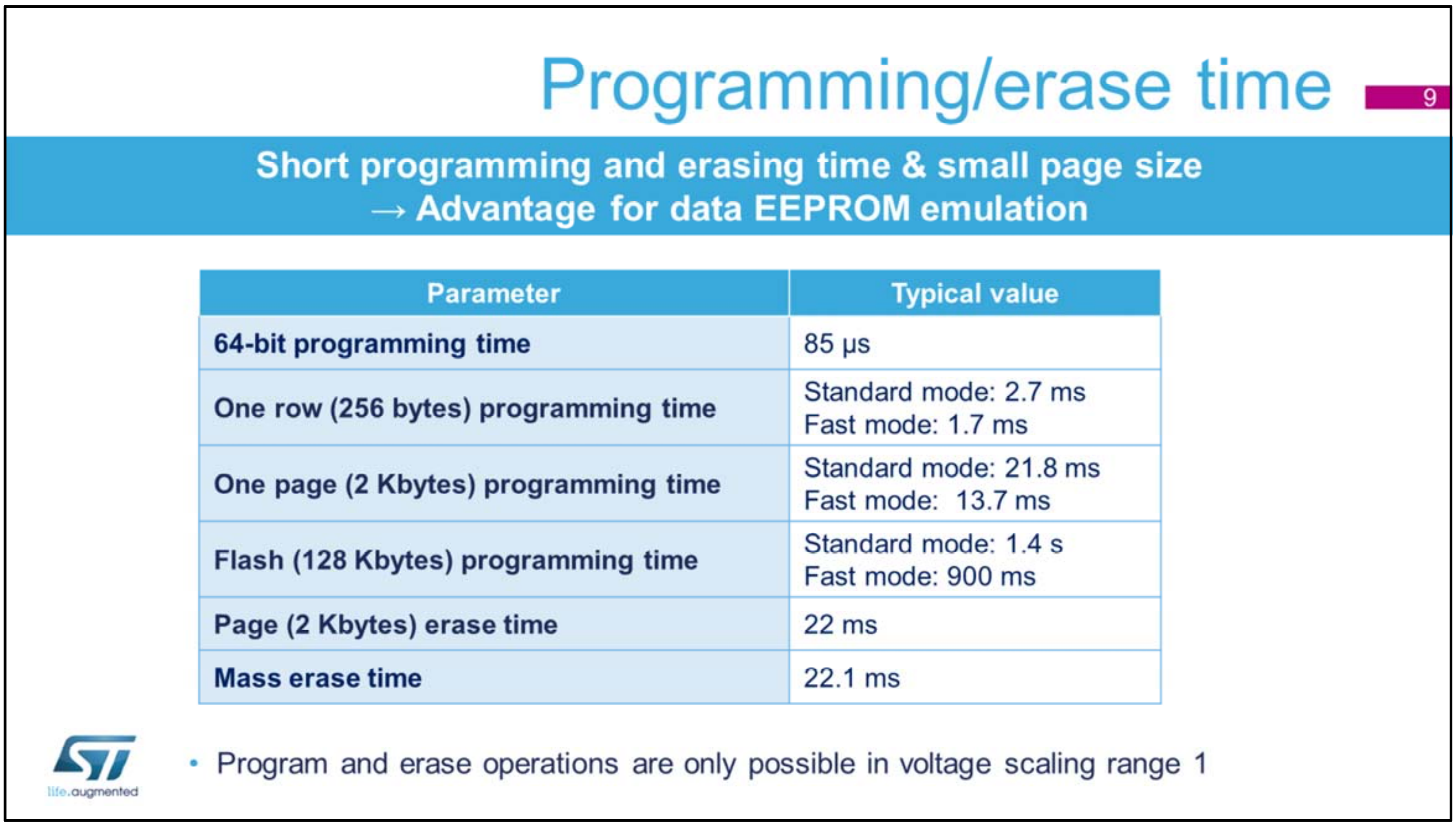
Flash memory features (2/2)

Robust memory integrity and safety

- 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 memory location

life.augmented

Fast programming enables the programming of a row of 256 bytes while normal programming has a granularity of 8 bytes.
The main purpose of Fast Programming is to reduce the page programming time. It is achieved by eliminating the need for verifying the Flash memory locations before they are programmed, thus saving the time of high-voltage ramping and falling for each double-word.



Programming/erase time

9

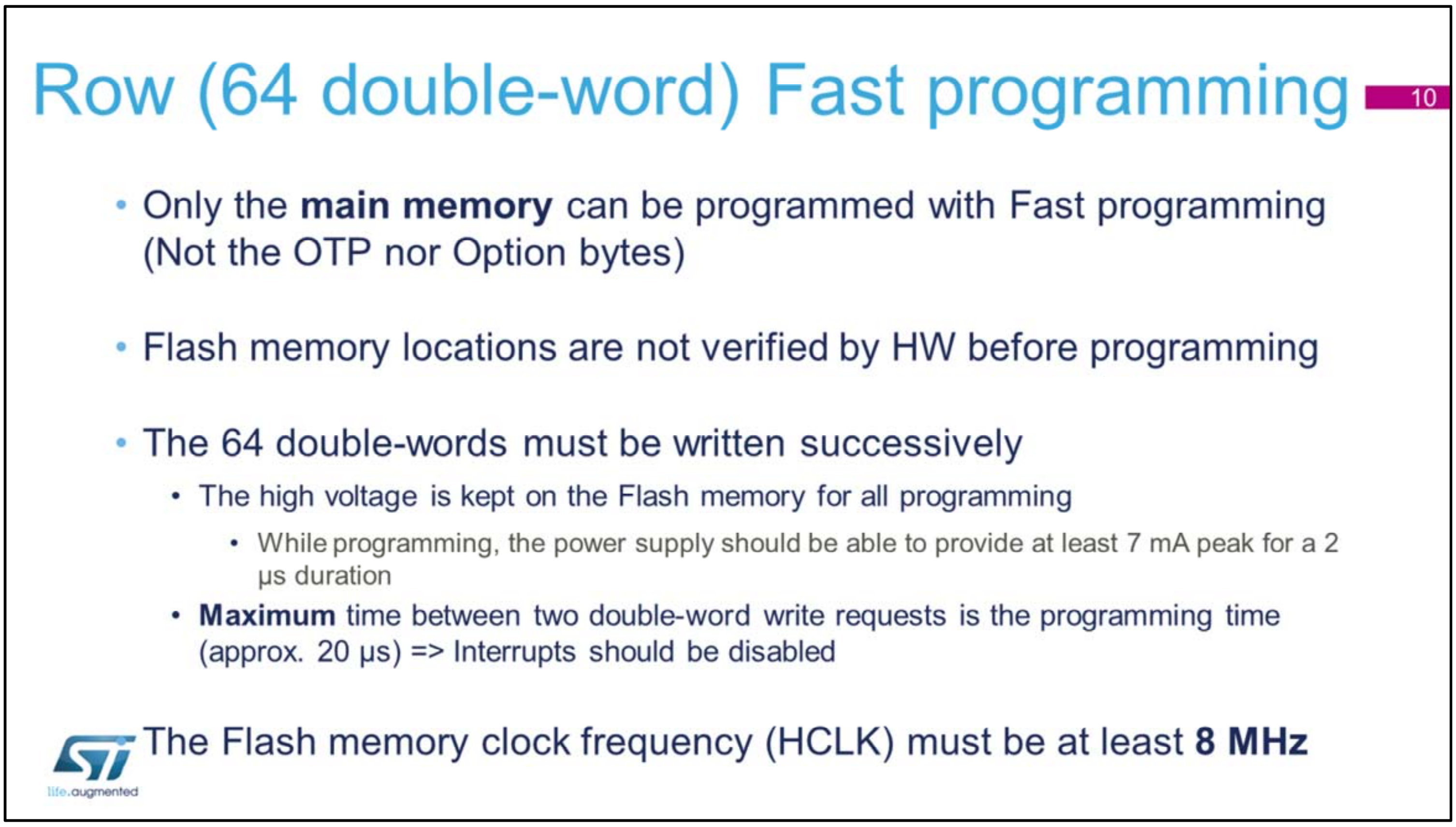
Short programming and erasing time & small page size
→ Advantage for data EEPROM emulation

Parameter	Typical value
64-bit programming time	85 µs
One row (256 bytes) programming time	Standard mode: 2.7 ms Fast mode: 1.7 ms
One page (2 Kbytes) programming time	Standard mode: 21.8 ms Fast mode: 13.7 ms
Flash (128 Kbytes) programming time	Standard mode: 1.4 s Fast mode: 900 ms
Page (2 Kbytes) erase time	22 ms
Mass erase time	22.1 ms

- Program and erase operations are only possible in voltage scaling range 1

life.augmented

Fast programming is 37% faster than standard mode programming.
Mass erase time, meaning a 128-Kbyte erase operation, approximately takes the same time as a page erase.



Row (64 double-word) Fast programming

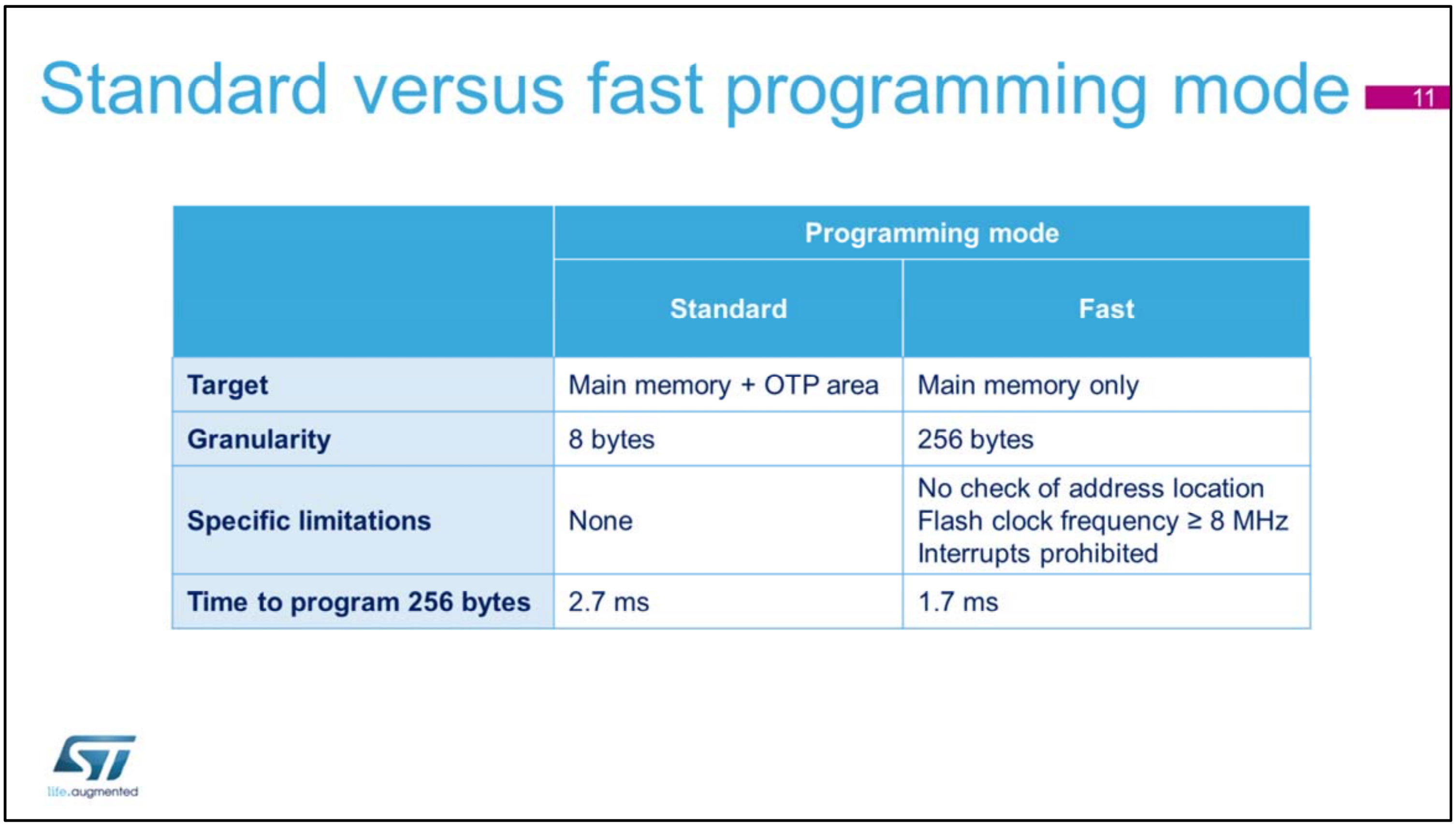
10

- Only the **main memory** can be programmed with Fast programming (Not the OTP nor Option bytes)
- Flash memory 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
    - While programming, the power supply should be able to provide at least 7 mA peak for a 2 µs duration
  - **Maximum** time between two double-word write requests is the programming time (approx. 20 µs) => Interrupts should be disabled
- The Flash memory clock frequency (HCLK) must be at least **8 MHz**

Fast programming vs standard programming:

- 256 consecutive bytes are programmed instead of 8-byte double-words located anywhere in the main Flash memory
- 8-byte programming is more reliable due to the verification step.

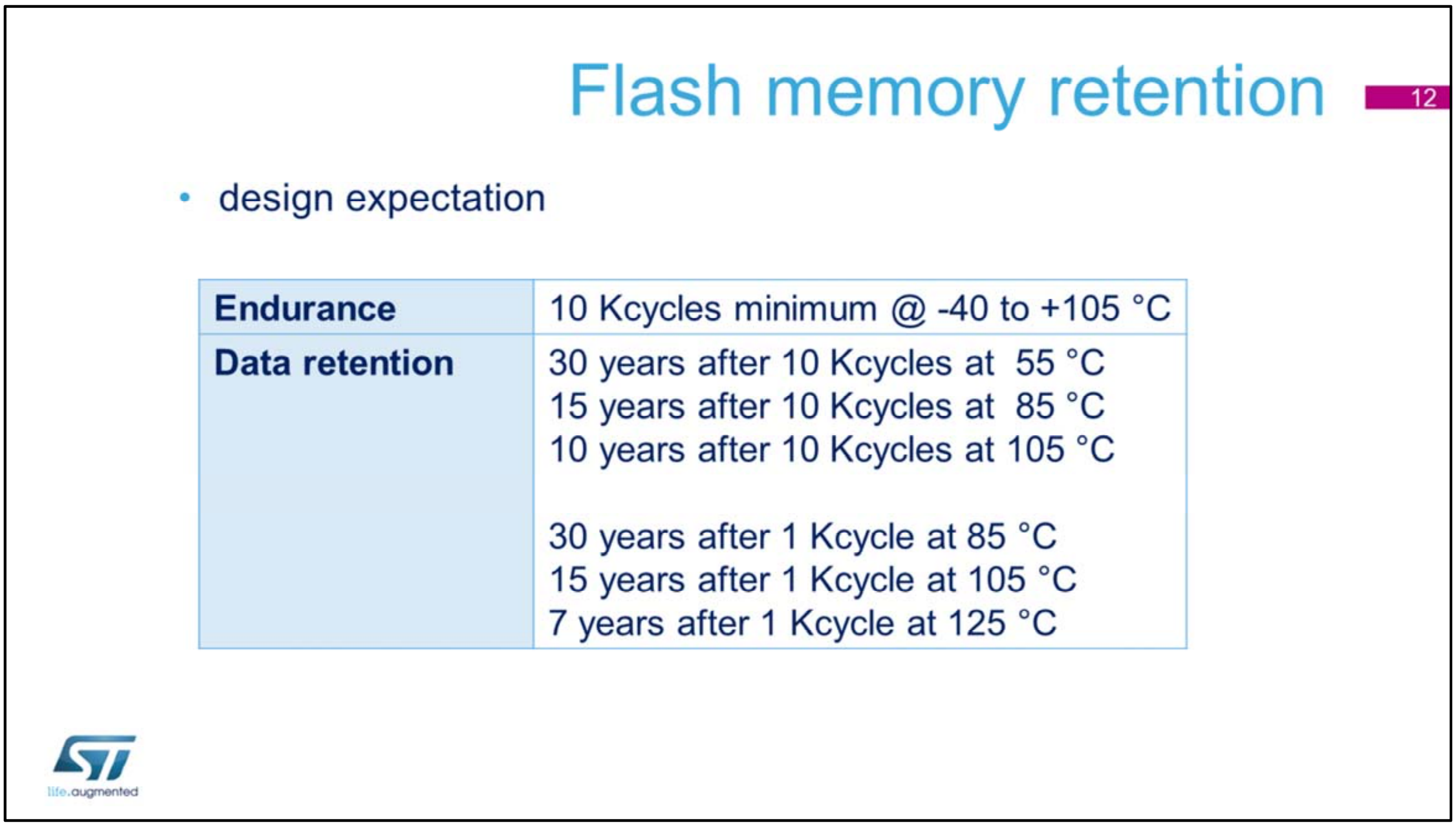
Note that the maximum time between two consecutive double words is around 20 µs. If a second double word arrives after this delay, fast programming is aborted and a flag is set. Consequently interrupts should be disabled to make sure that this delay is not exceeded.



Standard versus fast programming mode

	Programming mode	
	Standard	Fast
Target	Main memory + OTP area	Main memory only
Granularity	8 bytes	256 bytes
Specific limitations	None	No check of address location Flash clock frequency ≥ 8 MHz Interrupts prohibited
Time to program 256 bytes	2.7 ms	1.7 ms

This table summarizes the differences between standard and fast programming.



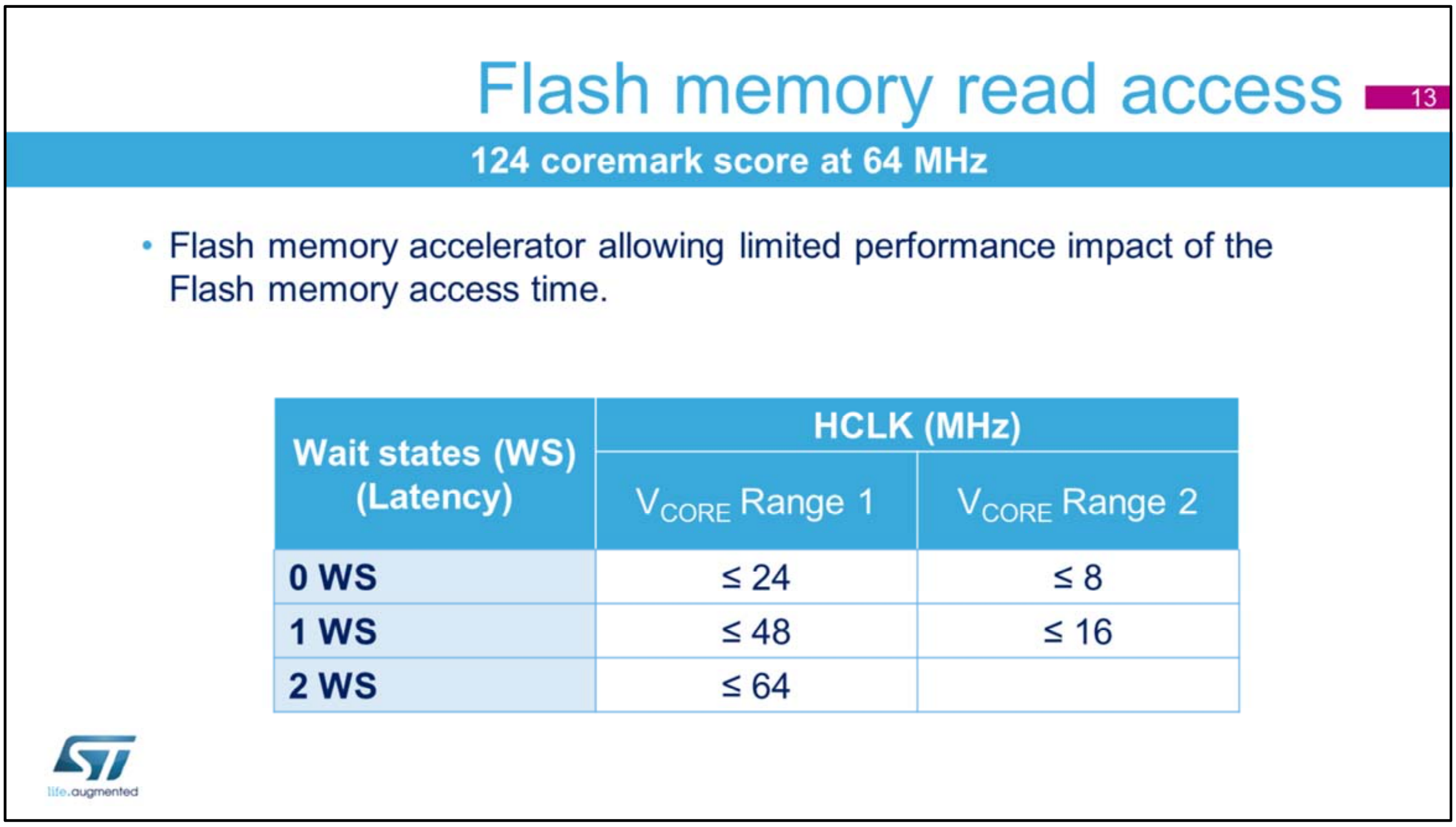
Flash memory retention

- design expectation

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

life.augmented

Each program / erase operation can degrade the Flash memory cell. After an accumulation of program / erase cycles, memory cells can become non-functional, causing memory errors.
Endurance is the maximum number of erase/programming sequences that the Flash memory can support without affecting its reliability.
Data retention is defined as retaining a given data pattern for a given amount of time.
The retention depends on the number of program/erase cycles and also on the temperature.



Flash memory read access

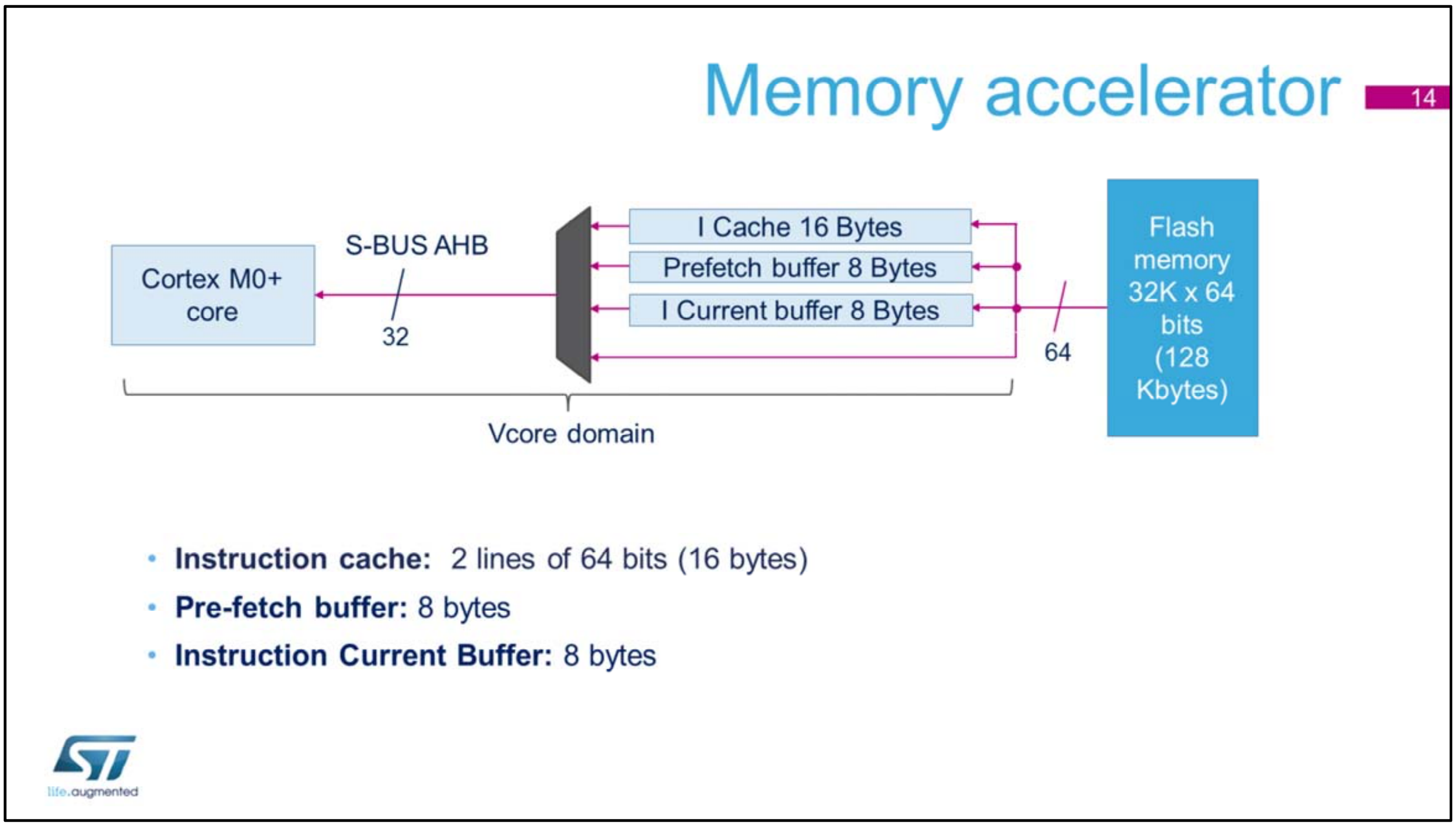
124 coremark score at 64 MHz

- Flash memory accelerator allowing limited performance impact of the Flash memory access time.

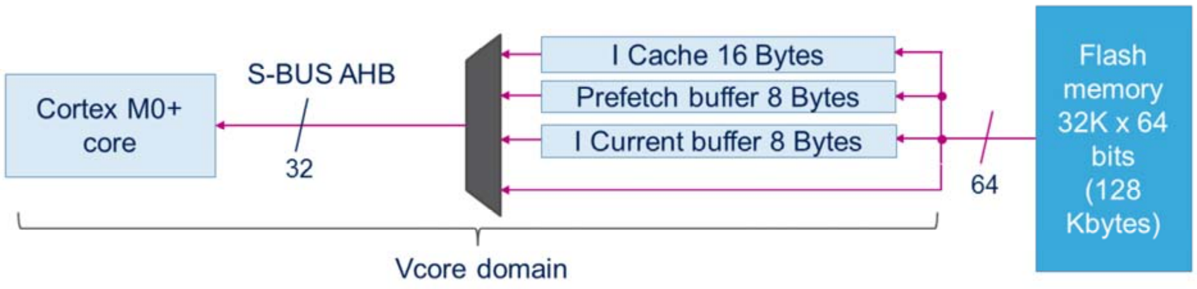
Wait states (WS) (Latency)	HCLK (MHz)	
	V_{CORE} Range 1	V_{CORE} Range 2
0 WS	≤ 24	≤ 8
1 WS	≤ 48	≤ 16
2 WS	≤ 64	

life.augmented

The Flash memory has a fixed access time while the AHB bus frequency can be dynamically change.
That is why the number of wait states is programmable and has to be set according to the actual AHB frequency, called HCLK.
Running beyond 64 MHz also requires to set the voltage scale to range 1. See the power controller (PWR) presentation.
Software is in charge of adjusting the number of wait states according to the HCLK frequency.
Increasing the number of wait states must be done prior to increasing the frequency.
Decreasing the number of wait states must be done after having decreased the frequency.
When the number of wait states is non null, the Flash memory accelerator should be activated to limit the performance impact.



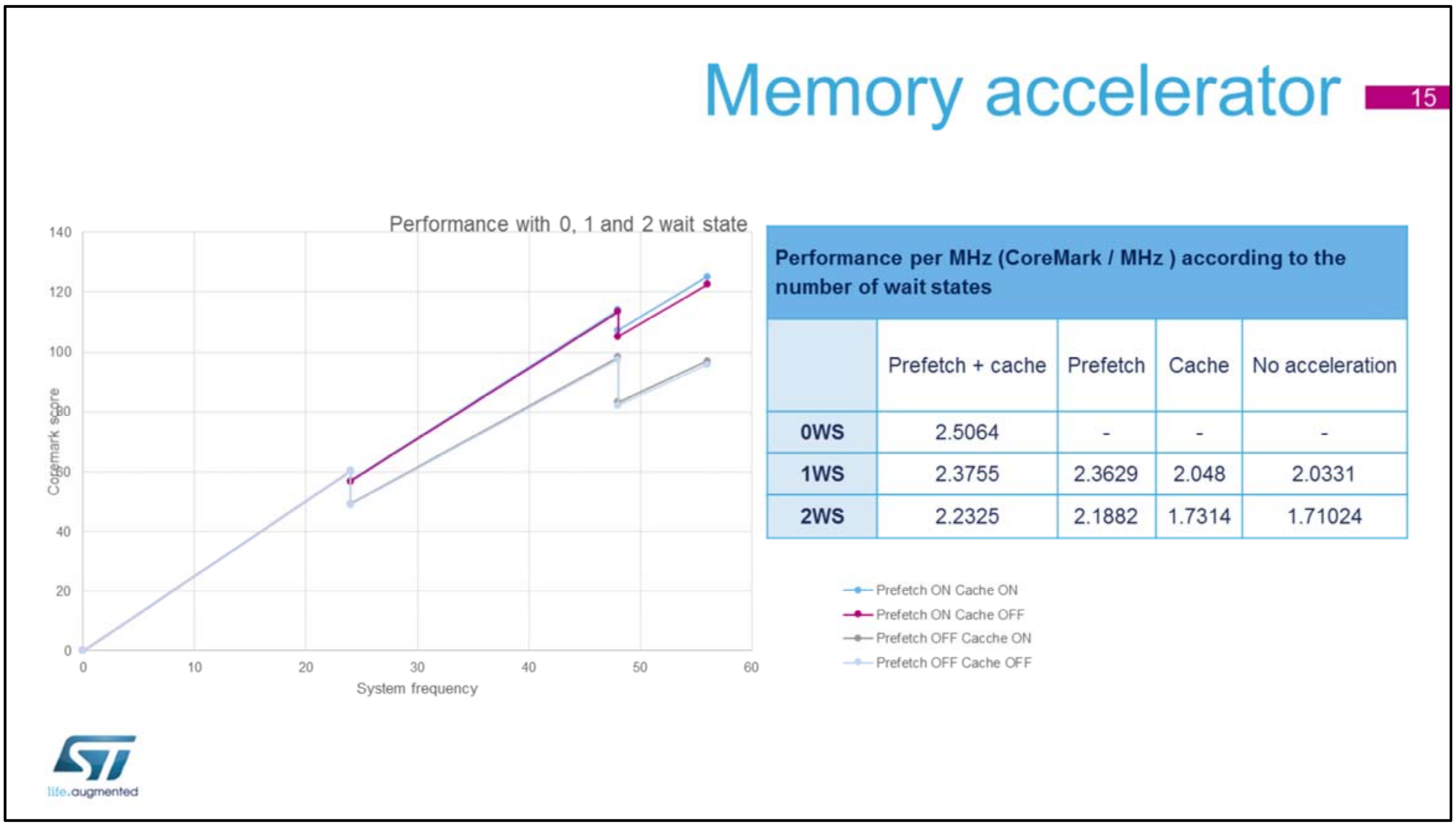
Memory accelerator



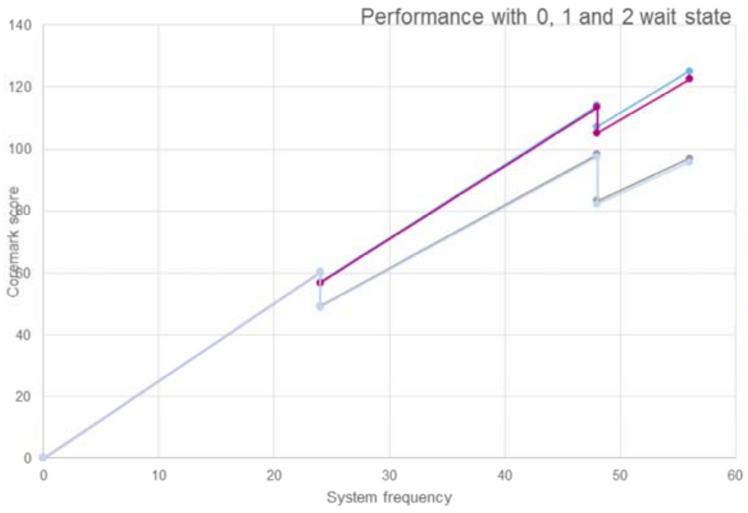
- **Instruction cache:** 2 lines of 64 bits (16 bytes)
- **Pre-fetch buffer:** 8 bytes
- **Instruction Current Buffer:** 8 bytes

CPU generates 32-bit instruction fetch requests. The 8-byte line containing the requested instruction is read from Flash memory and stored into the current buffer while the requested word is directly transferred to the CPU.
The next line is automatically read from Flash memory and stored into the prefetch buffer.
So, in case of sequential code, back-to-back words will be delivered over the S-AHB until a branch is encountered.
When the code is not sequential due to a branch, the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case, the penalty in terms of number of cycles is at least equal to the number of wait states.
Small loops can be entirely stored in the current and prefetch buffer, no Flash memory access is needed.

The Flash memory controller also implements an instruction cache of 16 bytes. Each time the requested instruction is not in the current and prefetch buffers, the line is copied into the instruction cache. If an instruction contained in the instruction cache memory is requested by the CPU, it is provided without inserting any delay. Once all the instruction cache memory lines are 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 in case of code containing loops. Instructions at the branch target address will be present in the instruction cache.
Both the prefetch buffer and instruction cache are enabled/disabled by software, because their impact on performance depends on the number of wait states to access the Flash memory. The instruction cache can also be reset by software.

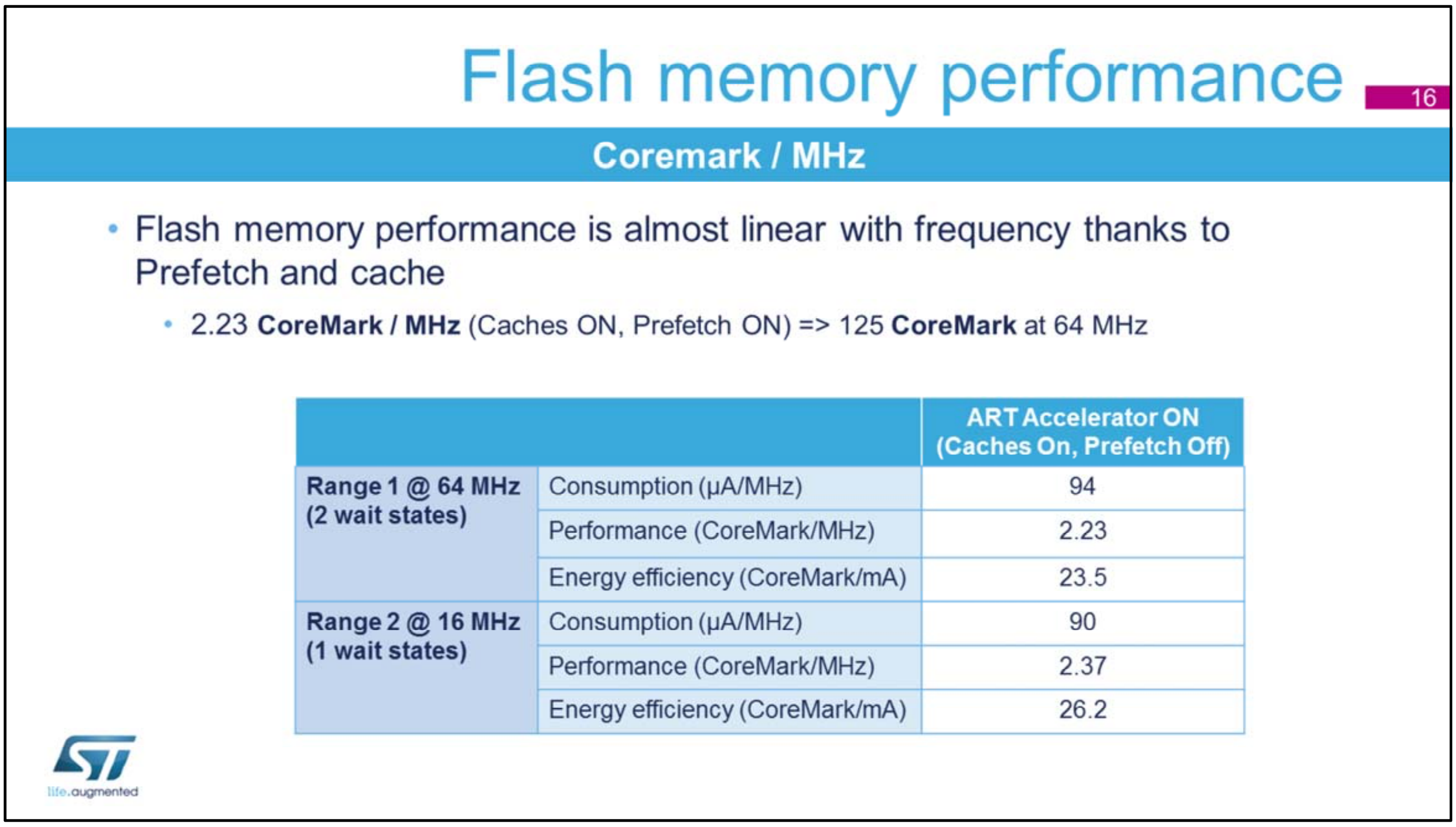


Memory accelerator



Performance per MHz (CoreMark / MHz) according to the number of wait states				
	Prefetch + cache	Prefetch	Cache	No acceleration
0WS	2.5064	-	-	-
1WS	2.3755	2.3629	2.048	2.0331
2WS	2.2325	2.1882	1.7314	1.71024

The performance continues to increase linearly with the frequency when accelerators are enabled (prefetch buffer and instruction cache).
The slope of the curve related to prefetch ON and cache ON is almost not affected by the transitions from 0 to 1 wait states achieved at 24 MHz and from 1 to 2 wait states achieved at 48 MHz.
From 0 to 24 MHz, enabling the prefetch buffer and the instruction cache does not improve the performance.



Flash memory performance

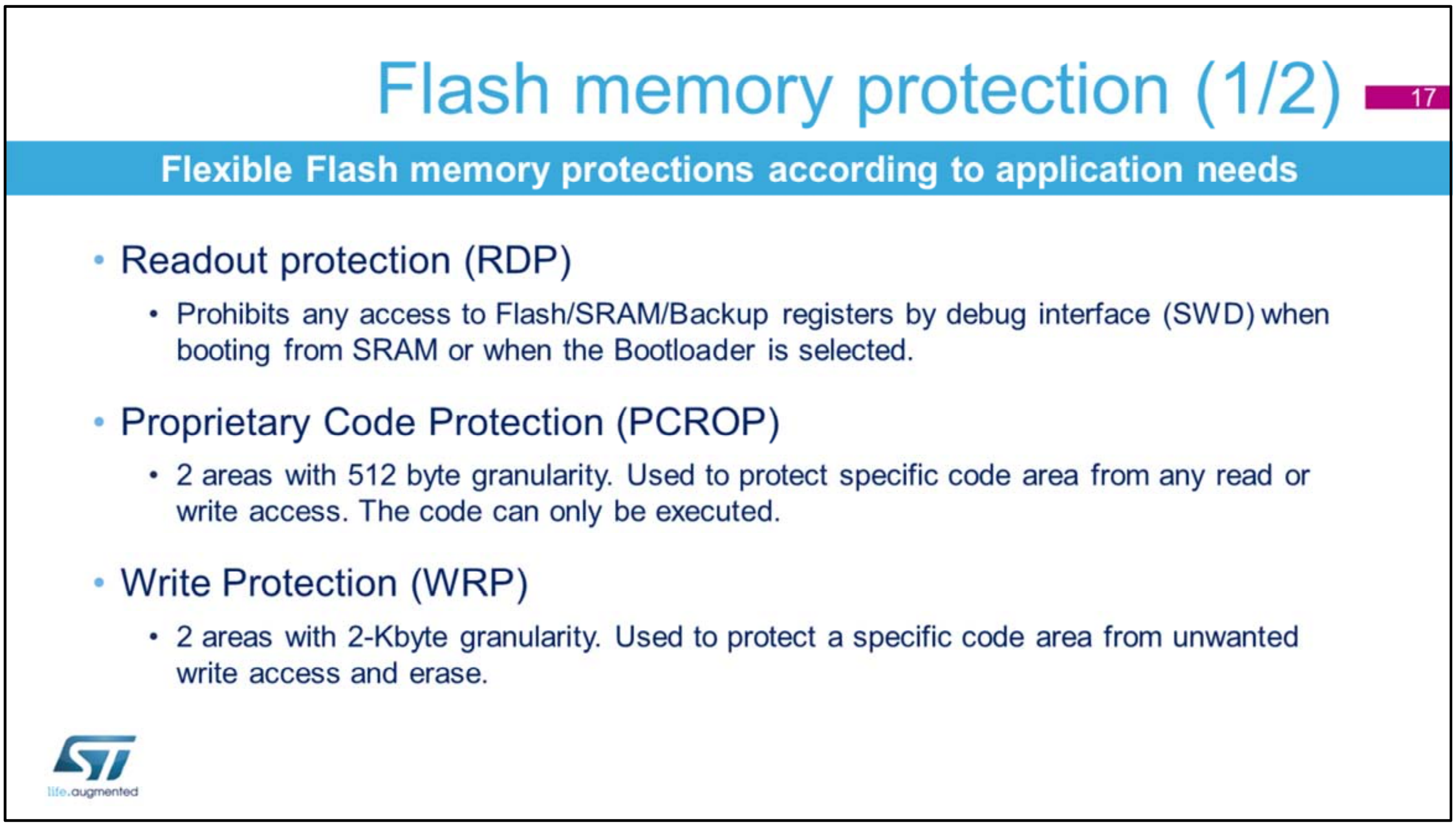
Coremark / MHz

- Flash memory performance is almost linear with frequency thanks to Prefetch and cache
  - 2.23 **CoreMark / MHz** (Caches ON, Prefetch ON) => 125 **CoreMark** at 64 MHz

		ART Accelerator ON (Caches On, Prefetch Off)
Range 1 @ 64 MHz (2 wait states)	Consumption (µA/MHz)	94
	Performance (CoreMark/MHz)	2.23
	Energy efficiency (CoreMark/mA)	23.5
Range 2 @ 16 MHz (1 wait states)	Consumption (µA/MHz)	90
	Performance (CoreMark/MHz)	2.37
	Energy efficiency (CoreMark/mA)	26.2

life.augmented

This array also shows that enabling the prefetch buffer and the instruction cache contributes to reducing consumption due to Flash memory accesses.
The consumption is only 4 microamps per MHz larger when running in power scale range 1 at a frequency of 64 MHz.
The reason is that prefetch buffer and instruction cache are located in the Vcore domain. When they provide the requested instruction, no Flash memory access is needed, which saves energy.



Flash memory protection (1/2)

17

Flexible Flash memory protections according to application needs

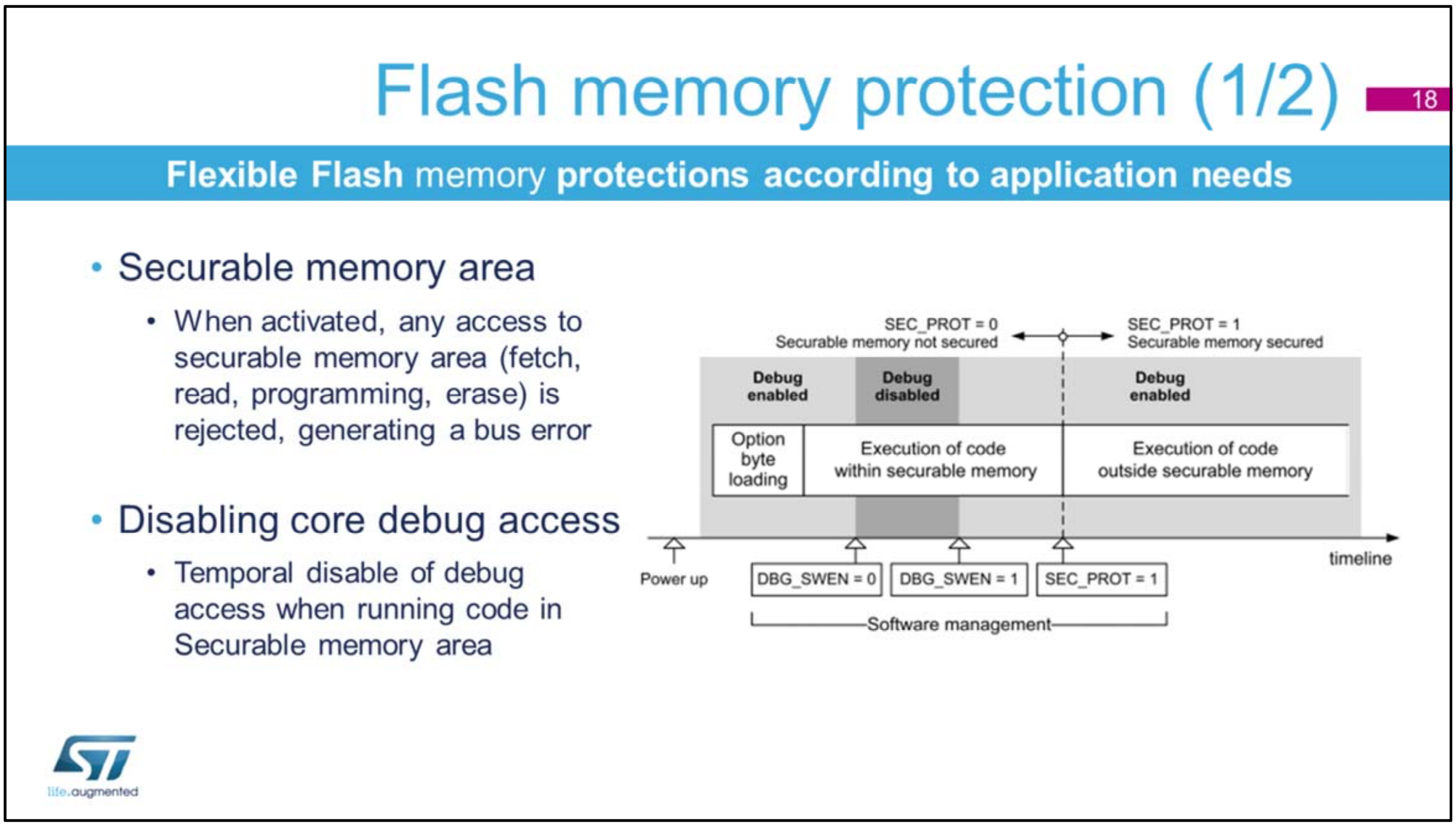
- Readout protection (RDP)
  - Prohibits any access to Flash/SRAM/Backup registers by debug interface (SWD) when booting from SRAM or when the Bootloader is selected.
- Proprietary Code Protection (PCROP)
  - 2 areas with 512 byte granularity. Used to protect specific code area from any read or write access. The code can only be executed.
- Write Protection (WRP)
  - 2 areas with 2-Kbyte granularity. Used to protect a specific code area from unwanted write access and erase.

life.augmented

Readout protection aims to protect the contents of the Flash memory, option bytes, internal SRAM and backup registers against reads requested by debuggers or software reads caused by programs executed after a boot from SRAM or bootloader.
Only a boot from Flash memory is permitted to read the contents of these memories.

The Proprietary Code Protection is a way to mark parts of the Flash memory as execute only. Note that this kind of access permissions is not supported by the Memory Protection Unit present in the Cortex®-M0+.
The user can declare two PCROP areas aligned on 512-byte addresses. PCROP areas are useful when only a part of the Flash memory has to be protected against third party reads
Write protection prevents part of the Flash memory from

being erased and reprogrammed.

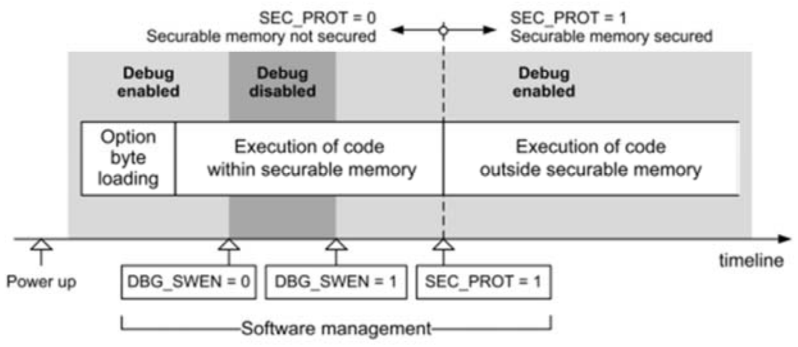


Flash memory protection (1/2)

18

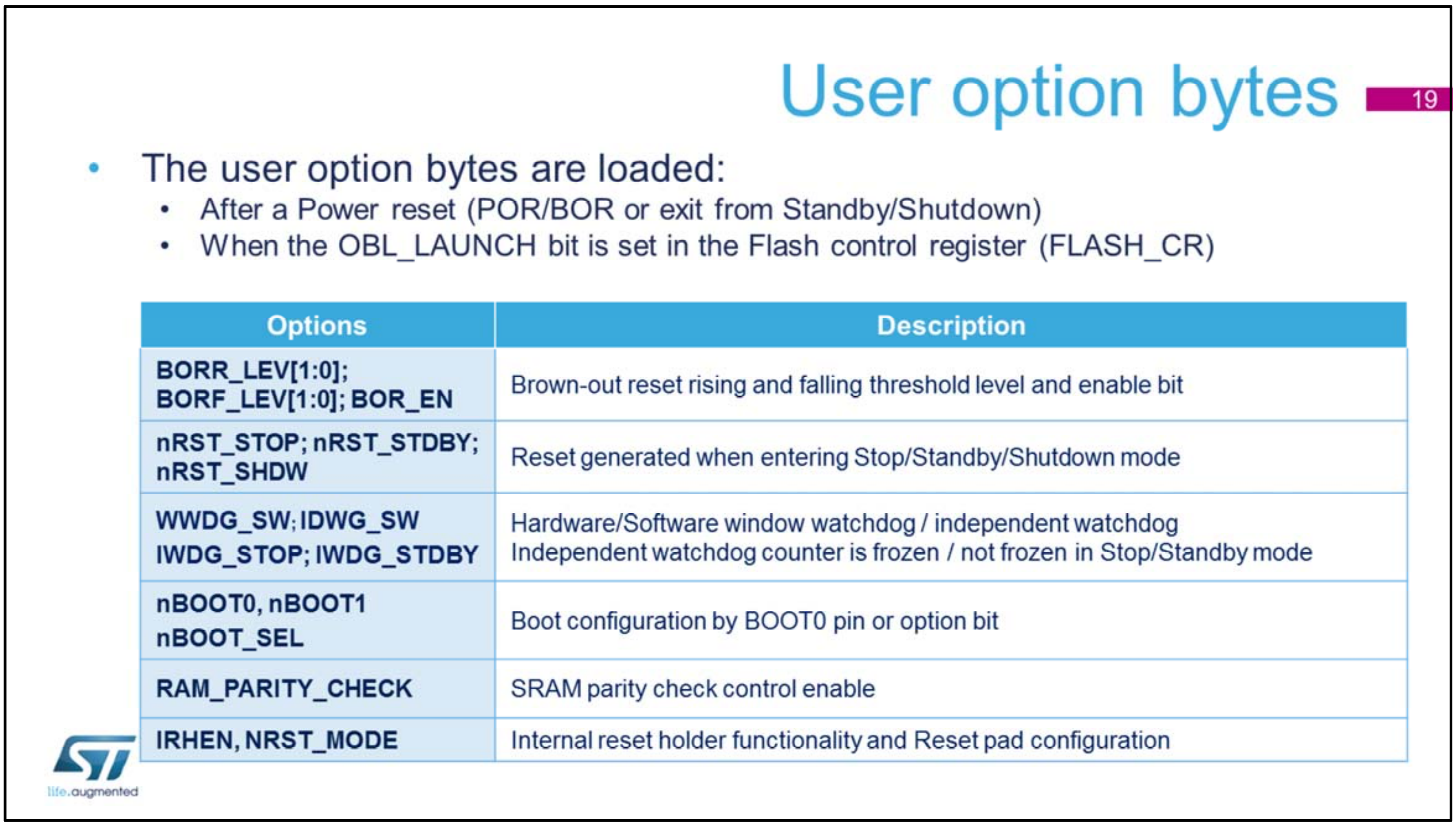
Flexible Flash memory protections according to application needs

- Securable memory area
  - When activated, any access to securable memory area (fetch, read, programming, erase) is rejected, generating a bus error
- Disabling core debug access
  - Temporal disable of debug access when running code in Securable memory area



life.augmented

The main purpose of the securable memory area is to protect a specific part of Flash memory against undesired access. This allows implementing software security services such as secure key storage or secure boot, in charge of image authentication.
Once the processor has exited the securable memory, this part of the Flash memory is no longer accessible.
The securable area can only be unsecured by a reset of the device.
The size of the securable memory area is aligned on 2-Kbyte pages.
In addition, the code executed from the securable memory can temporarily disable debug accesses.



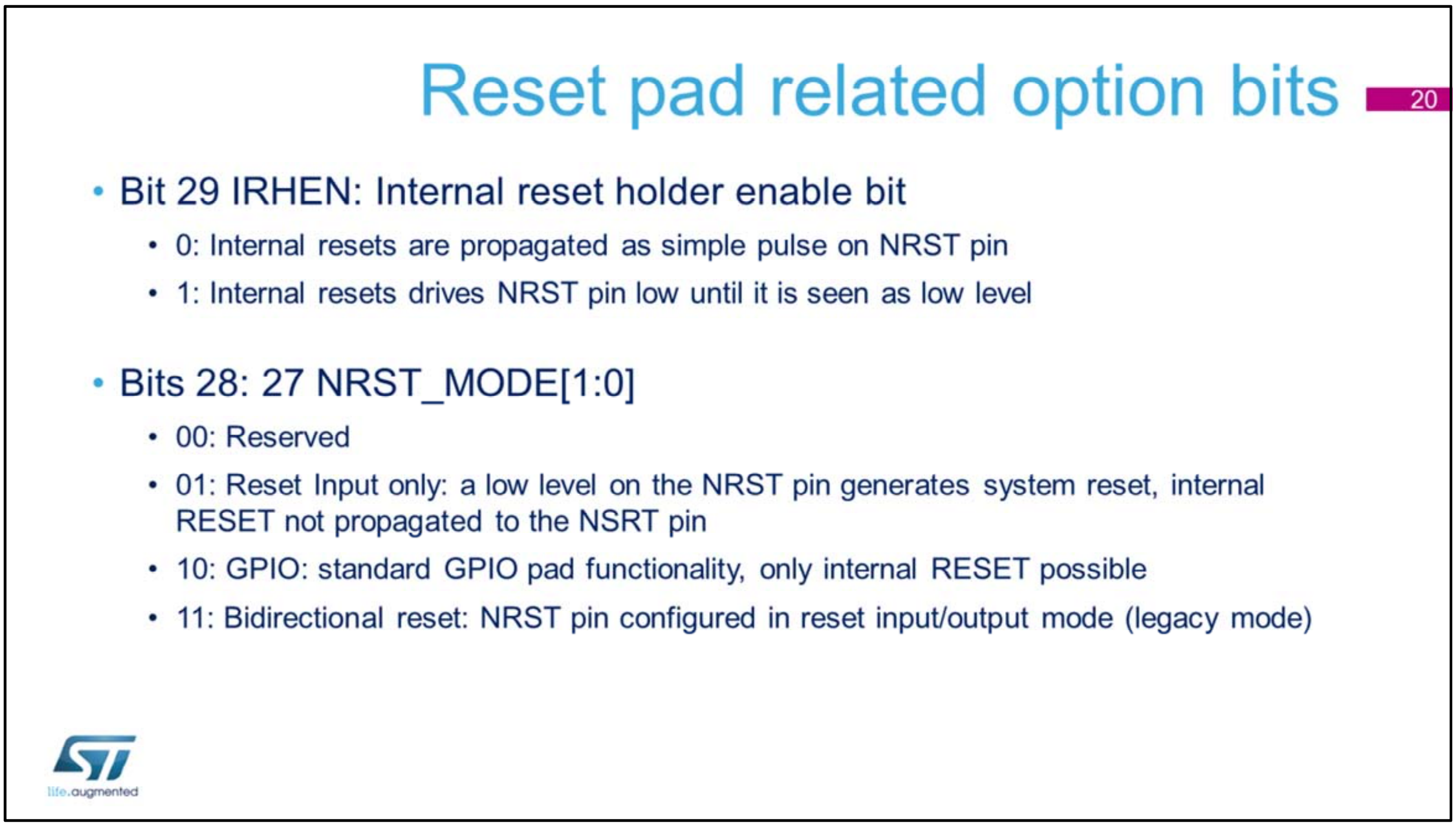
User option bytes

- The user option bytes are loaded:
  - After a Power reset (POR/BOR or exit from Standby/Shutdown)
  - When the OBL_LAUNCH bit is set in the Flash control register (FLASH_CR)

Options	Description
BORR_LEV[1:0]; BORF_LEV[1:0]; BOR_EN	Brown-out reset rising and falling threshold level and enable bit
nRST_STOP; nRST_STDBY; nRST_SHDW	Reset generated when entering Stop/Standby/Shutdown 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
nBOOT0, nBOOT1 nBOOT_SEL	Boot configuration by BOOT0 pin or option bit
RAM_PARITY_CHECK	SRAM parity check control enable
IRHEN, NRST_MODE	Internal reset holder functionality and Reset pad configuration

life.augmented

Option Bytes are used to early configure the system-on-chip before starting the Cortex®-M0+. They represent 128 Bytes.
They are automatically loaded after a power reset or on request by setting the OBL_LAUNCH bit in the FLASH_CR register. This capability is required to apply a new setting without resetting the device.
This slide and the two next ones describe the various fields of the Option Bytes.



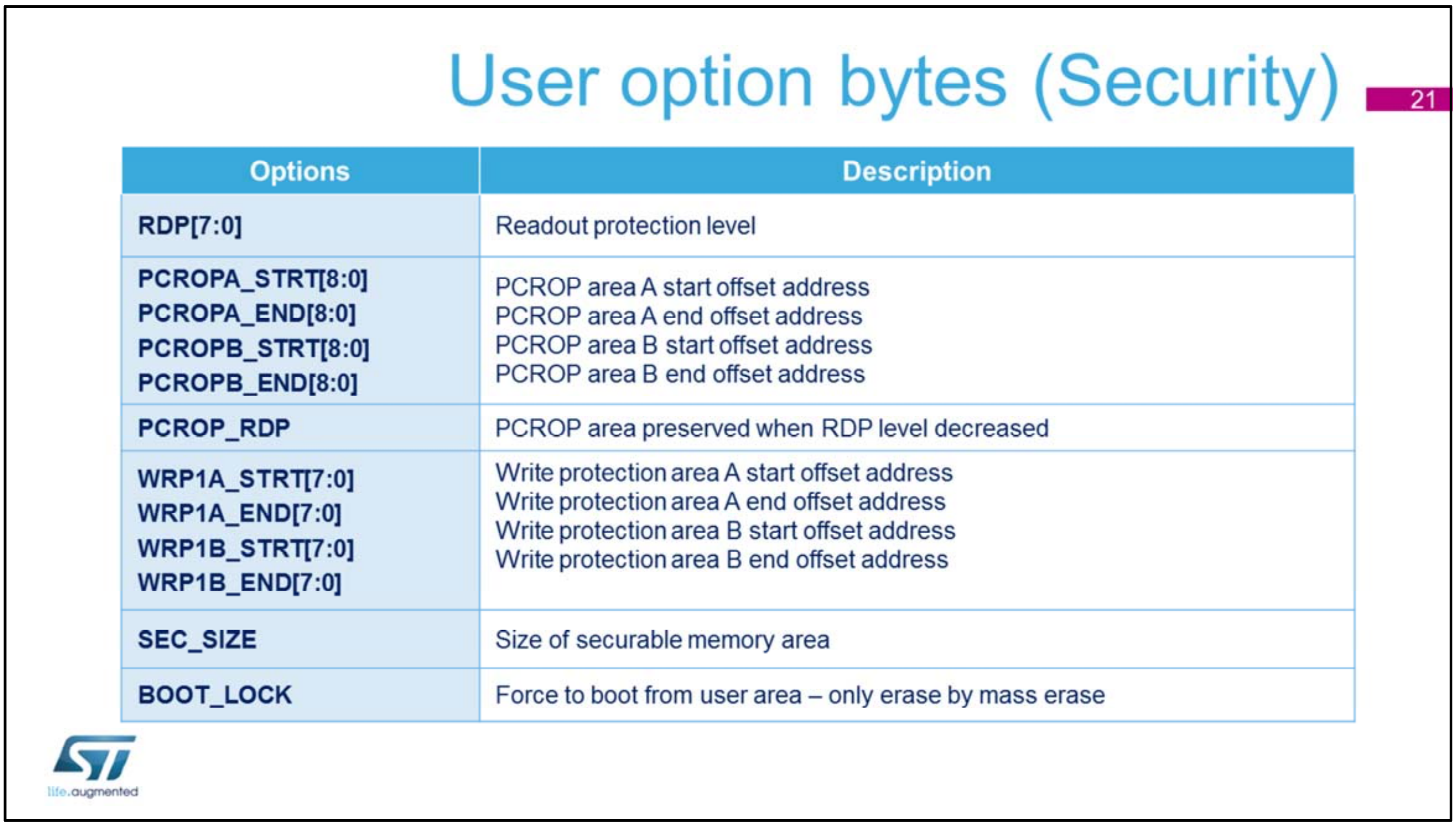
Reset pad related option bits

20

- Bit 29 IRHEN: Internal reset holder enable bit
  - 0: Internal resets are propagated as simple pulse on NRST pin
  - 1: Internal resets drives NRST pin low until it is seen as low level
- Bits 28: 27 NRST_MODE[1:0]
  - 00: Reserved
  - 01: Reset Input only: a low level on the NRST pin generates system reset, internal RESET not propagated to the NSRT pin
  - 10: GPIO: standard GPIO pad functionality, only internal RESET possible
  - 11: Bidirectional reset: NRST pin configured in reset input/output mode (legacy mode)

life.augmented

Bit 28 configures the NRST pin: either as a GPIO, as a reset input only or as a reset input and output.
When it is a reset input and output, bit 29 configures the output stage: either a pulse generator or a low level driver which drives the pin low until it is seen as low level. This is useful when the reset line has an important capacitive load.



User option bytes (Security)

Options	Description
RDP[7:0]	Readout protection level
PCROPA_STRT[8:0] PCROPA_END[8:0] PCROPB_STRT[8:0] PCROPB_END[8:0]	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_STRT[7:0] WRP1A_END[7:0] WRP1B_STRT[7:0] WRP1B_END[7:0]	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
SEC_SIZE	Size of securable memory area
BOOT_LOCK	Force to boot from user area – only erase by mass erase

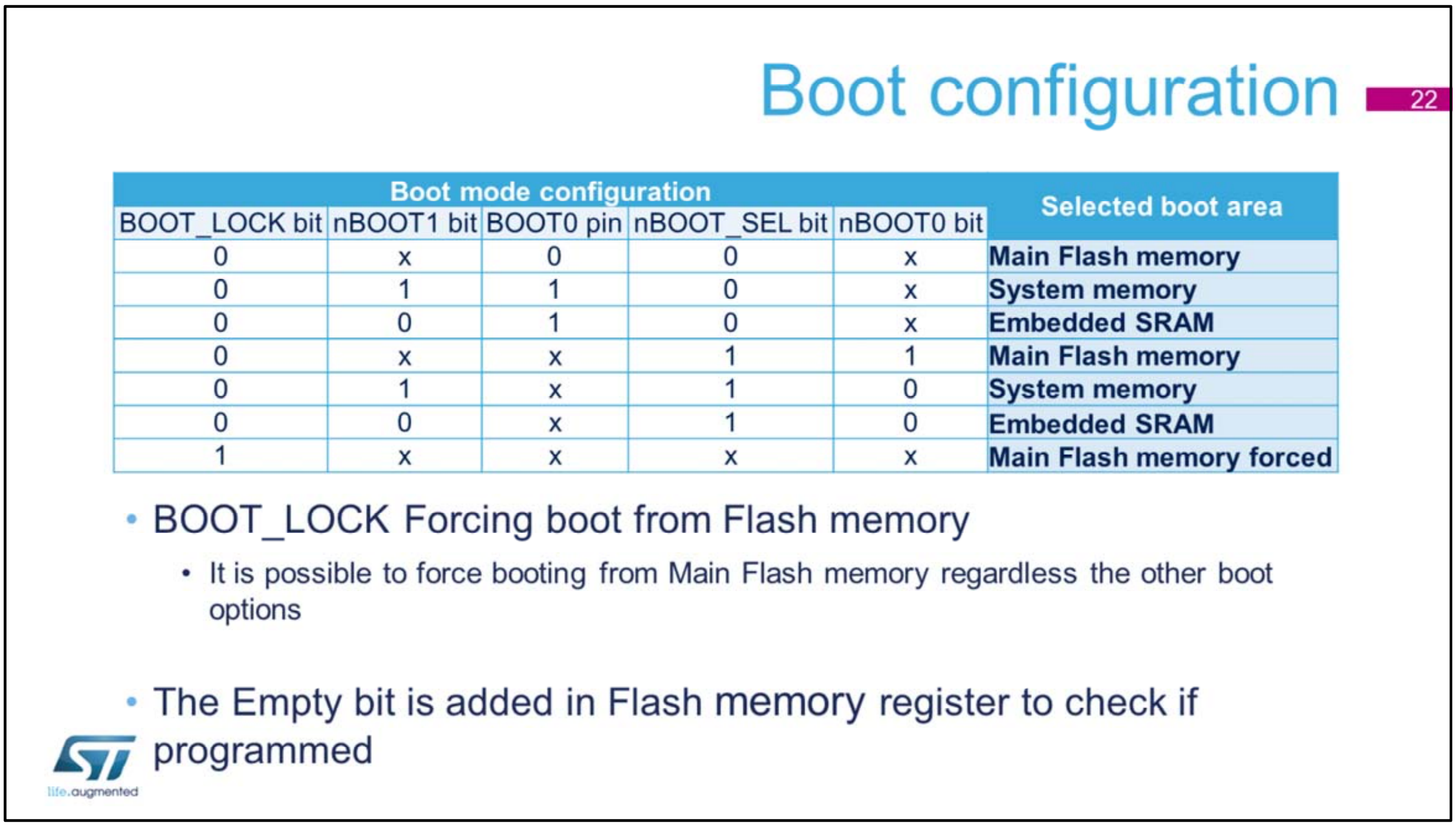
The readout protection level enables the readout protection for the entire Flash memory:

- Level 0: no protection
- Level 1: read protection
- Level 2: no debug.

The following transitions are supported: Level 0 to Level 1, Level 1 to Level 0 which implies a partial or mass erase, Level 0 to Level 2 and Level 1 to Level 2.

- PCROPA_STRT and PCROPA_END define the proprietary code readout protection address range A aligned on 512 bytes.
- PCROPB_STRT and PCROPB_END define the proprietary code readout protection address range B aligned on 512 bytes.
- PCROP_RDP allows to select if the PCROP area is erased or not when the RDP protection is changed from Level 1 to Level 0.

- SEC_SIZE defines the size of the securable memory.
- Boot_lock allows forcing the system to boot from the Main Flash memory regardless the other boot options.

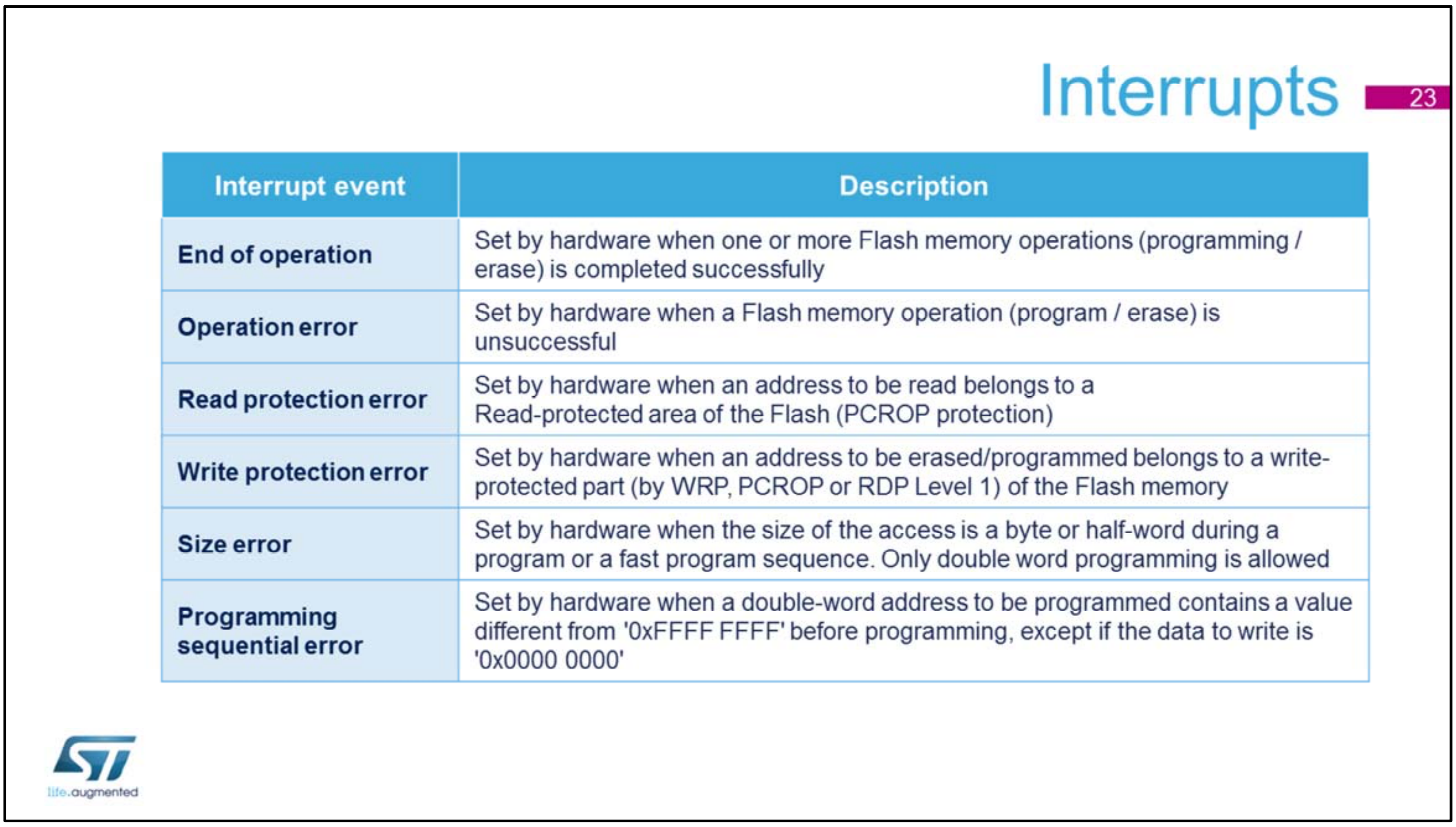


Boot configuration

Boot mode configuration					Selected boot area
BOOT_LOCK bit	nBOOT1 bit	BOOT0 pin	nBOOT_SEL bit	nBOOT0 bit	
0	x	0	0	x	Main Flash memory
0	1	1	0	x	System memory
0	0	1	0	x	Embedded SRAM
0	x	x	1	1	Main Flash memory
0	1	x	1	0	System memory
0	0	x	1	0	Embedded SRAM
1	x	x	x	x	Main Flash memory forced

- BOOT_LOCK Forcing boot from Flash memory
  - It is possible to force booting from Main Flash memory regardless the other boot options
- The Empty bit is added in Flash memory register to check if programmed

The boot memory is selected from both option bytes and also from the BOOT0 pin. This table indicates in which memory the processor will boot according to the combination of parameters.
Note that when nBOOT_SEL bit is set to 1, the BOOT0 pin is ignored. Only option bytes select the boot memory.
When the BOOT_LOCK bit is set in option bytes, only boot from Flash memory is supported.
During the Option bytes Loading phase, after loading all options, the Flash memory interface checks whether the first location of the Main memory is programmed. The result of this check in conjunction with the Boot 0 and Boot 1 information is used to determine where the system has to boot from. It prevents the system to boot from Main Flash memory area when no user code has been programmed.

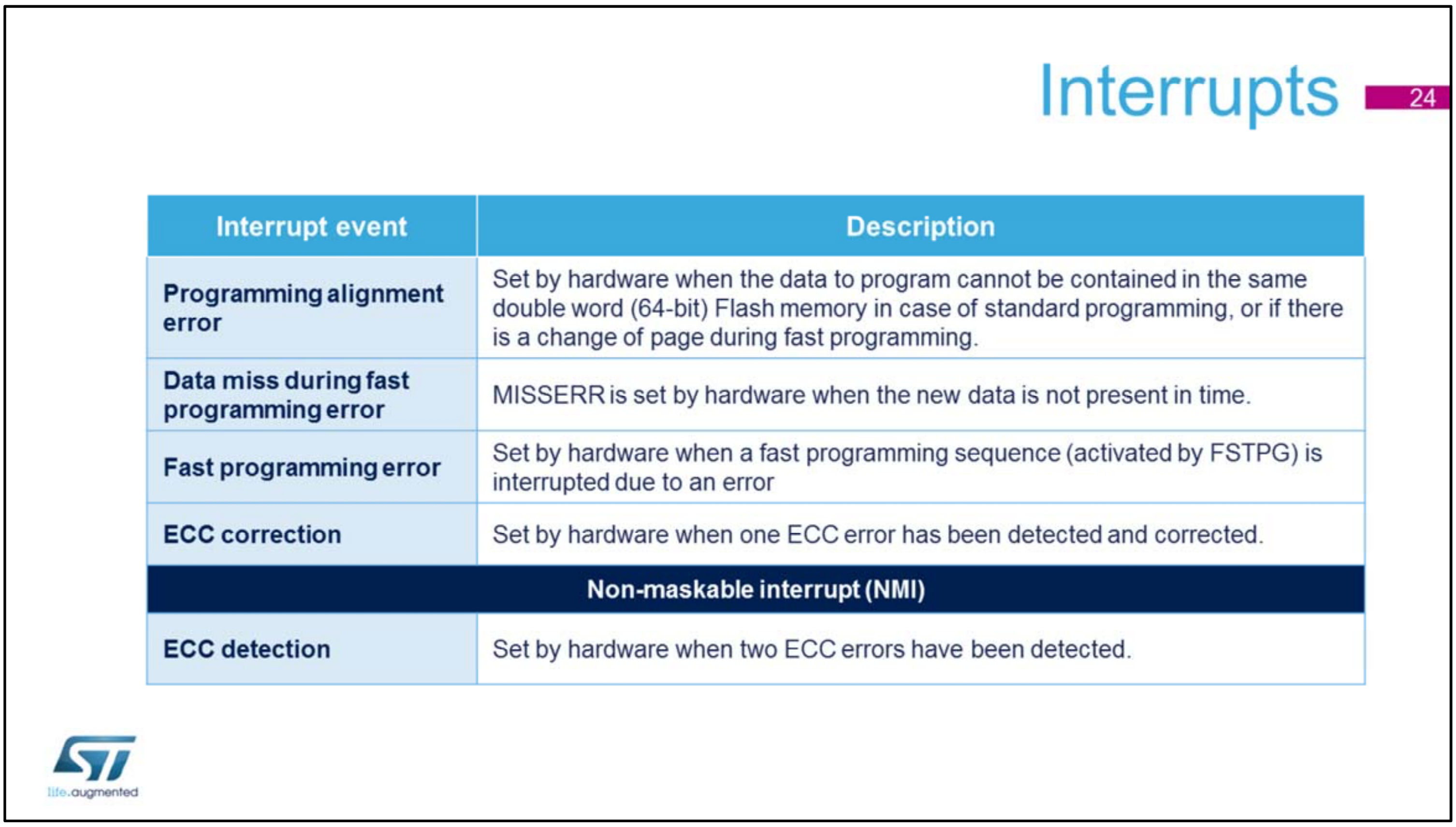


Interrupts

Interrupt event	Description
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 protection error	Set by hardware when an address to be read belongs to a Read-protected area of the Flash (PCROP protection)
Write protection error	Set by hardware when an address to be erased/programmed belongs to a write-protected part (by WRP, PCROP or RDP Level 1) of the Flash memory
Size error	Set by hardware when the size of the access is a byte or half-word during a program or a fast program sequence. Only double word programming is allowed
Programming sequential error	Set by hardware when a double-word address to be programmed contains a value different from '0xFFFF FFFF' before programming, except if the data to write is '0x0000 0000'

life.augmented

The Flash memory controller supports many interrupt sources, listed in this slide and the next one.
An interrupt can be asserted upon successful end of operation.
An interrupt can also be asserted when an error occurs during a program / erase operation.
Protection violations can also cause interrupts.
A Size error occurs when the data to be programmed is not word-aligned.
Programming sequential error occurs when a program operation is attempted without having previously erased the location in Flash memory.

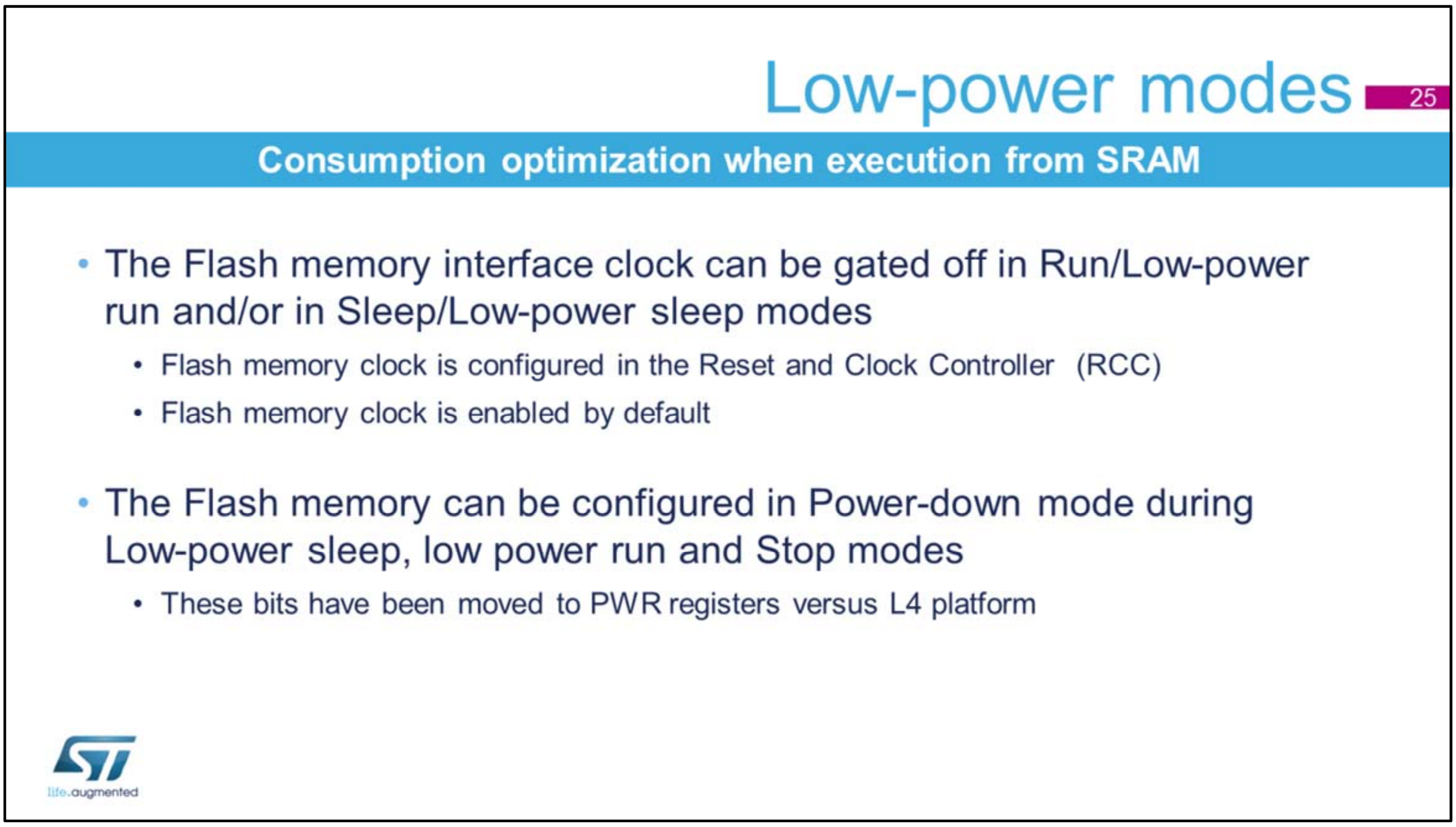


Interrupts

Interrupt event	Description
Programming alignment error	Set by hardware when the data to program cannot be contained in the same double word (64-bit) Flash memory in case of standard programming, or if there is a change of page during fast programming.
Data miss during fast programming error	MISSERR is set by hardware when the new data is not present in time.
Fast programming error	Set by hardware when a fast programming sequence (activated by FSTPG) is interrupted due to an error
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

A programming alignment error occurs when a complete double word is not provided before initiating a standard program operation or when a complete row is not written before initiating a fast programming operation.
A Data miss programming error occurs when data are not written in time during a fast programming sequence.
When a single-bit ECC error is detected and fixed, an interrupt can be asserted.
When a double-bit ECC error is detected, the NMI is asserted.

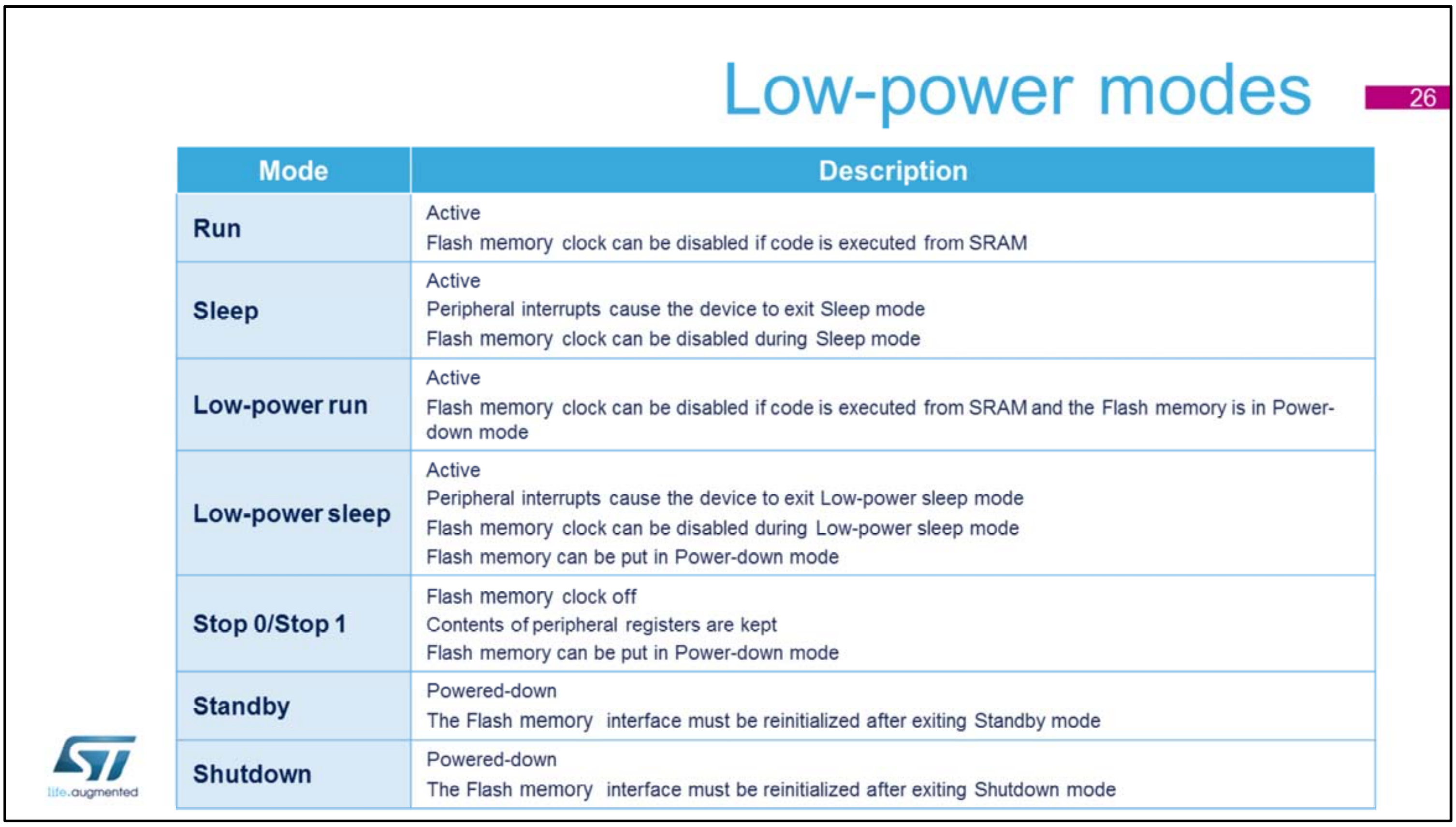


Low-power modes

Consumption optimization when execution from SRAM

- The Flash memory interface clock can be gated off in Run/Low-power run and/or in Sleep/Low-power sleep modes
  - Flash memory clock is configured in the Reset and Clock Controller (RCC)
  - Flash memory clock is enabled by default
- The Flash memory can be configured in Power-down mode during Low-power sleep, low power run and Stop modes
  - These bits have been moved to PWR registers versus L4 platform

The Flash memory module can be clock-gated when the processor does not need to access the Flash memory and also in low-power modes.
The Flash memory module can also be power-gated in Sleep, Run and Stop modes.



Low-power modes

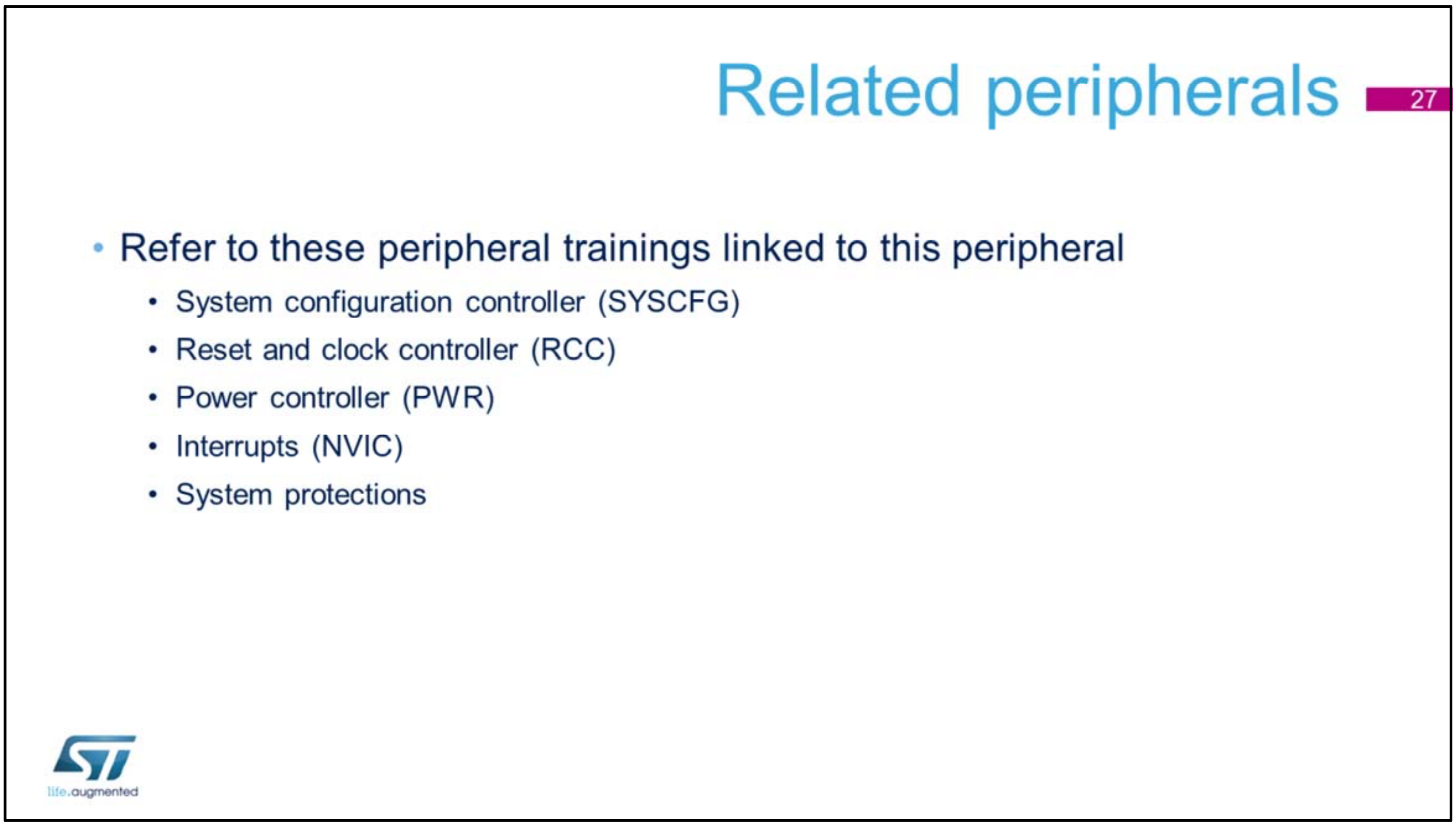
Mode	Description
Run	Active Flash memory clock can be disabled if code is executed from SRAM
Sleep	Active Peripheral interrupts cause the device to exit Sleep mode Flash memory clock can be disabled during Sleep mode
Low-power run	Active Flash memory clock can be disabled if code is executed from SRAM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

life.augmented

The Flash memory module supports the following low power capabilities:

- Clock gating
- Flash memory power-down mode
- Power gating of the entire module: Flash memory and controller.

In Run and Sleep modes, only clock gating is supported.
In Low-power Run and Low-power Sleep modes, the Flash memory can enter Power-down mode while the clock of the controller is gated.
In Stop0 and Stop1, the clocks are gated and Flash memory can enter Power-down mode.
In Shutdown mode, the power of the Flash memory module is gated, for both the Flash memory and controller.

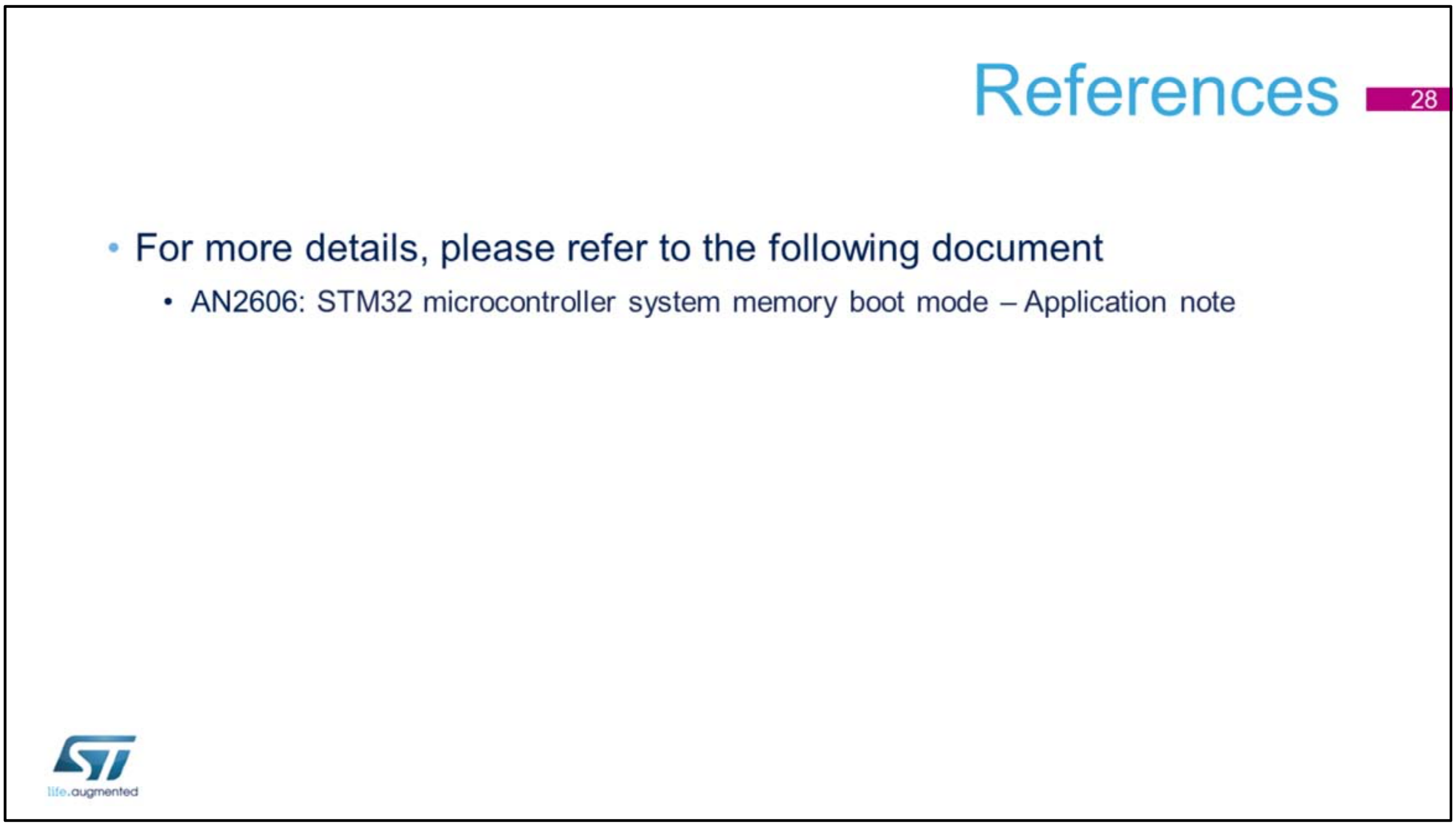


Related peripherals

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

The Flash memory module has relationships with the following other modules:

- System configuration controller (SYSCFG)
- Reset and clock controller (RCC)
- Power controller (PWR)
- Interrupts (NVIC)
- System protections.



References

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

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