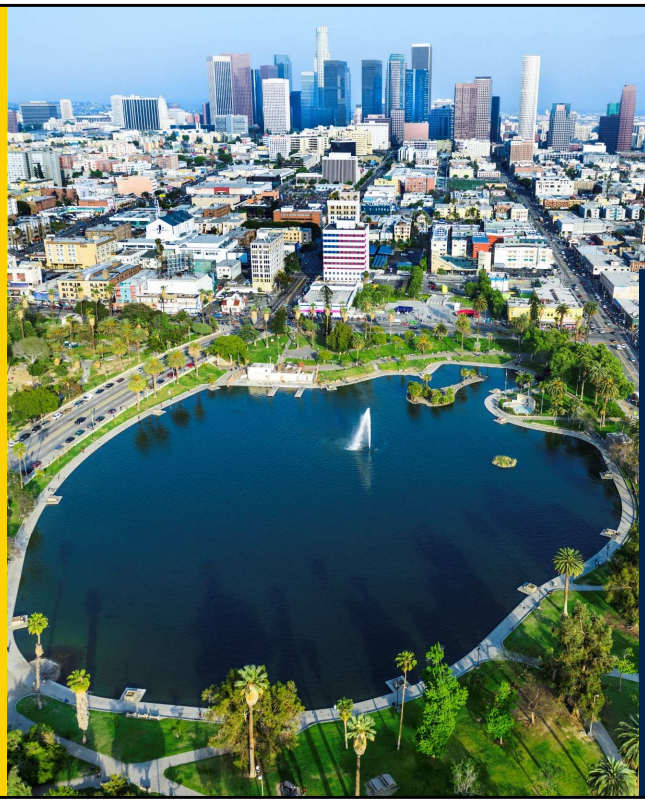


life.augmented

STM32U0 – FLASH Memory Flash

Hello, and welcome to this presentation of the embedded Flash memory which is included in all products of the STM32U0 microcontroller family.

Main differences with STM32L0

- The Flash memory interface is much different from the one implemented in STM32L0 microcontrollers
- This table lists the main differences with the STM32L0

	STM32L0	STM32U0
Instruction Cache	No	1 KB
Prefetch	32bit	2x32bit
OTP Area	No	1 Kbyte
ECC redundancy	Implicit	With output
Securable Memory	No	Yes
EEPROM for data	Yes	No*

life.augmented

The STM32U0's Flash memory interface supports new features with respect to the STM32L0, 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.
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

- STM32U0 embeds up to 256 Kbytes of single-bank Flash memory
- The Flash memory interface manages all access (read, programming, erasing), memory protection, security and option byte programming
- 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

Application benefits

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

The STM32U0 embeds up to to 256 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 of 2 KB and short programming time.
It provides various security and protection mechanisms for code and data, read and write access.

Key features

- Up to 256 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

The STM32U0 embeds up to 32 Kbytes of flash memory on STM32U031xx devices and up to 256 Kbytes (32 bytes x 72 bits) on STM32U073xx and STM32U083xx devices.
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 a 16-byte prefetch buffer and 1-KB instruction cache.
They also contribute 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 128 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

5

In addition to the 256 Kbytes of the main Flash memory, the STM32U0 supports:

- A System memory of 26 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
	…	…	…
	0x0803_F800 – 0x0803_FFFF	2 Kbytes	Page 127
Information block	0x1FFF_0000 – 0x1FFF_67FF	26 Kbytes	System memory
	0x1FFF_6800 – 0x1FFF_6BFF	1 Kbyte	OTP area
	0x1FFF_6C00 – 0x1FFF_7FFF	5 Kbytes	Non-user area

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

life.augmented

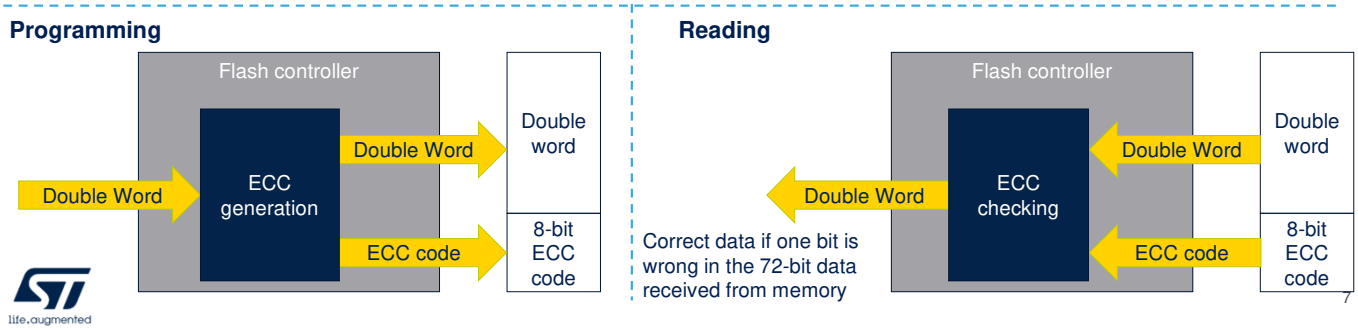
The first table details the memory organization based on a Main Flash memory area and an information block.
The second table details the granularity of the Flash memory operations:

- 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.

Flash memory features (1/2)

Robust memory integrity and safety

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



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

- Single-bit error detection and correction
- Double-bit error 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.
When an ECC error is detected, the address of the failing double word is saved in FLASH_ECCR register.

Flash memory features (2/2)

Programming modes

- Programming granularity is 64 bits (really 72 bits including 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

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

life.augmented

9

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

- Only the **main memory** can be programmed with Fast programming (Neither the OTP nor Option bytes)
- Flash memory locations are not verified by hardware 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 therefore be disabled
- The Flash memory clock frequency (HCLK) must be at least **8 MHz**

life.augmented

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 an error 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

life.augmented

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

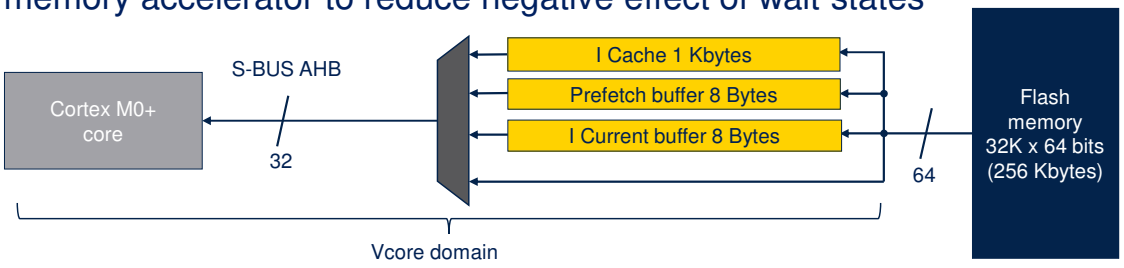
life.augmented

Each erase/program operation can degrade the Flash memory cell.
After an accumulation of erase/program cycles, memory cells can become non-functional, causing memory errors.
Endurance is the maximum number of erase/program 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 erase/program cycles and also on the temperature.

Flash memory read access

140 coremark score at 56 MHz

- Flash memory accelerator to reduce negative effect of wait states



Wait states (WS) (Latency)	HCLK (MHz)	
	V_{CORE} Range 1	V_{CORE} Range 2
0 WS	≤ 24	≤ 8
1 WS	≤ 48	≤ 16
2 WS	≤ 56	≤ 18

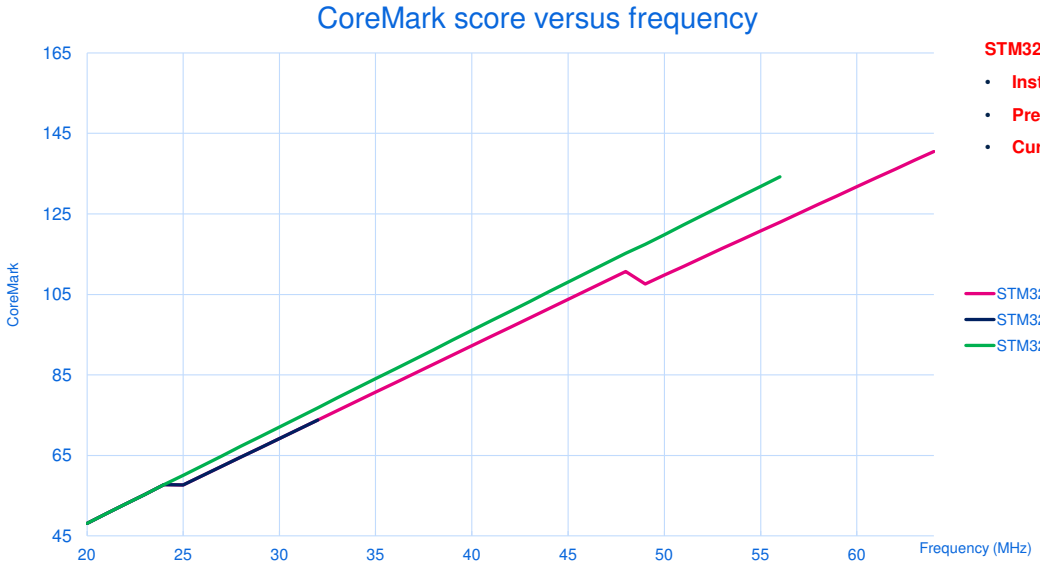
Instruction cache a and prefetch makes the execution more effective with higher clock speed

life.augmented

13

The Flash memory has a fixed access time while the AHB bus frequency can be dynamically changed.
That is why the number of wait states is programmable and has to be set according to the actual AHB frequency, called HCLK.
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 prefetch buffer and instruction cache should be activated to limit the performance impact.

Memory accelerator



STM32U0 series efficiency advantages

- **Instruction cache:** 1 Kbyte
- **Pre-fetch buffer:** 8 bytes
- **Current instruction buffer:** 8 bytes

life.augmented

14

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

		Instruction cache on, prefetch buffer 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

15

This array also shows that enabling the prefetch buffer and the instruction cache contributes to reducing consumption by minimizing the number of 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 (1/3)

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
- Write Protection (WRP)
  - 2 areas with 2-Kbyte granularity
    - Used to protect a specific code area from unwanted write access and erase

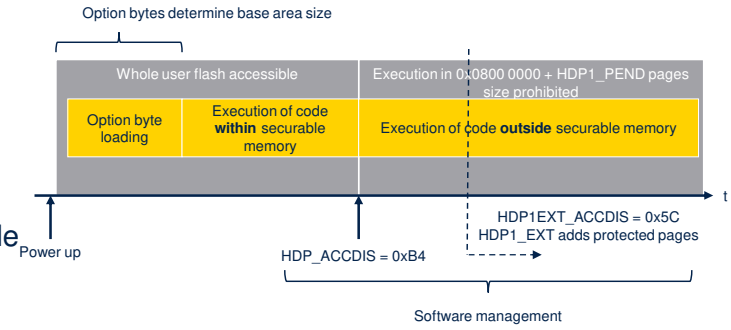
life.augmented

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.
Write protection prevents part of the Flash memory from being erased and reprogrammed.

Flash memory protection (2/3)

Flexible flash memory protections according to application needs

- Securable memory area
  - Configured in the OB
  - When activated, any access to securable memory area (fetch, read, programming, erase) is rejected, generating a bus error
- Securable memory area extension
  - Code can extend the reach of the securable memory area
  - Page granularity
  - Volatile setting, works one way



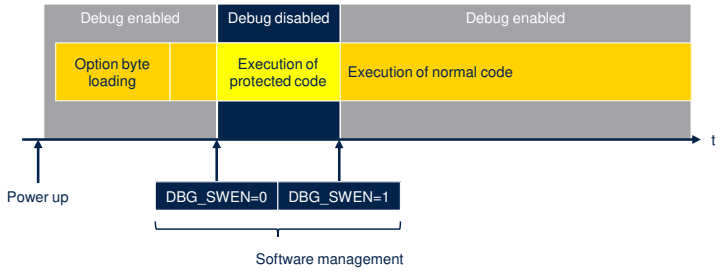
life.augmented

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.

Flash memory protection (3/3)

Flexible flash memory protections according to application needs

- Disabling core debug access
  - Temporal disable of debug access when running protected code, for example in securable memory area



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 (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 level
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
BDRST	Backup domain reset
BKPSRAM_HW_ERASE_DISABLE	Backup SRAM erase prevention

life.augmented

Option Bytes are used to early configure the system-on-chip before starting the Cortex®-M0+. They represent 96 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 next two describe the various fields of the Option Bytes.

User option bytes
Reset pad related option bits

Options	Description
IRHEN, NRST_MODE	Internal reset holder functionality and Reset pad configuration

- 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

Bits 28 and 27 configure 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
OEM1KEY[127:0]	Two 128-bit keys (OEM1KEY and OEM2KEY) can be defined and used to control the RDP regression
OEM2KEY[127:0]	
HDP1EN[7:0] HDP1_PEND[6:0]	Hide protection area enable and last page of hide protection area
BOOT_LOCK	Force to boot from user area – only erase by mass erase
WRP1A_STRT[6:0], WRP1A_END[6:0], WRP1B_STRT[6:0], WRP1B_END[6:0]	Write protection configuration, 2 areas

life.augmented

The readout protection (RDP) level enables the readout protection for the entire Flash memory:

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

The OEM1 RDP lock mechanism is active when the OEM1LOCK bit is set in the FLASH_SR register. It blocks the RDP level 1 to RDP level 0 regression. The unlock sequence is based on a comparison between OEM1KEY and the key passed in the DBGMCU_DBG_AUTH_HOST register.

The OEM2 RDP lock mechanism is active when the OEM2LOCK bit is set in the FLASH_CR register. It blocks the RDP level 2 to RDP level 1 regression. The unlock sequence is based on a comparison between OEM2KEY and the key passed in the DBGMCU_DBG_AUTH_HOST

register
The pages can also be protected against unwanted write (WRP) due to loss of program counter context. Two regions can be defined aligned on 2 Kbyte pages.
The Secure Hide Protection (HDP) protects a specific part of flash memory against undesired access. The securable memory area is located in the first pages of the main flash memory. The last page is defined by the HDP1_PEND[6:0] bitfield.
Secure state is active when HDP1_ACCDIS[7:0] value in the FLASH_HDPCR register is different from 0xA3, and the HDP1EN[7:0] value in FLASH_SECR option byte register is different from 0xB4.
Then, any write access (programming, erase) to the securable memory area is rejected and generates a bus error.
Boot_lock allows forcing the system to boot from the Main Flash memory regardless the other boot options.
However, it is possible to reset this bit only when:

- RDP is set to level 0, or
- RDP is set to level 1, while level 0 is requested and a full mass-erase is performed.

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

life.augmented

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 (1/2)

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
Write protection error	Set by hardware when an address to be erased/programmed belongs to a write-protected part (by WRP, 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
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

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.
If an erase/program operation to a write-protected part of the flash memory is attempted, the write protection error interrupt can be raised.
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.
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.

Interrupts (2/2)

Interrupt event	Description
Programming sequence error	Programming or erase sequence requirements not satisfied
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 sequence interrupt is raised in various situations when the proper programming or erase sequence procedure is not satisfied.
A Data miss programming error occurs when data are not written on time during a fast-programming sequence.
A fast programming error interrupt occurs if one of the following conditions occurs:

- When the Fast programming enable control bit is set for more than 8 ms, which generates a time-out detection
- When the row fast programming has been interrupted.

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, sleep, and 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 run, low power run, sleep, and low-power sleep run modes
  - Refer to the presentation on PWR module

life.augmented

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.
Refer to the presentations on RCC and PWR.
If a flash memory programming operation is ongoing, Stop, Standby, and Shutdown entry is delayed until the flash memory interface access is finished.

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/ Stop 2	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, low-power run, sleep and low-power sleep modes, both clock- and flash memory power-gating are supported.
In Stop0, Stop1, and Stop2, the clocks are gated, and flash memory can enter Power-down mode.
In Standby and Shutdown modes, 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)
  - Security- Memory protections (MEMPROTEC)

life.augmented

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)
- Security- Memory protections (MEMPROTEC).

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.

Thank you

life.augmented

Thanks for attending this presentation.