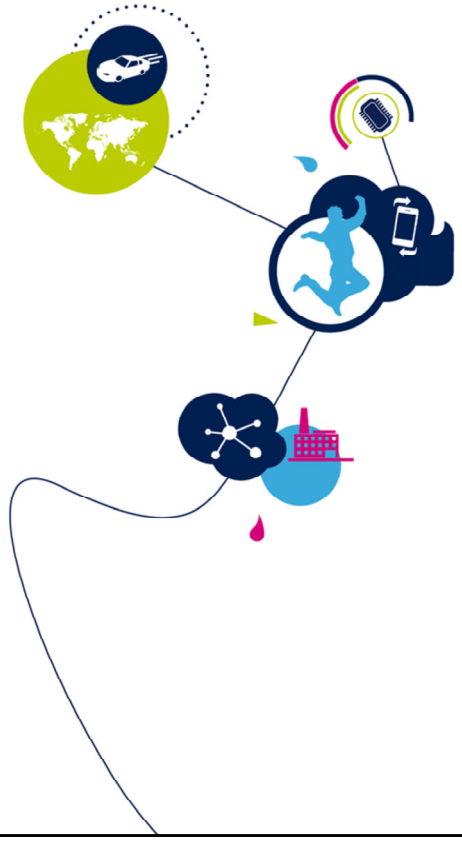
STM32G4 - Flash

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 STM32G4 microcontroller family.

Overview

- STM32G4 embeds up to 512 Kbytes of Flash memory with dual-bank architecture
- The Flash memory interface manages all access (read, programming, erasing), memory protection, security and option byte programming

Application benefits

- High-performance and low-power
- Read-while-write capability
- Small erase granularity
- Short programming time
- Dual-bank booting
- Security and protection

life.augmented

The STM32G4 microcontrollers embed up to 512 Kilobyte of Flash memory with dual-bank architecture. 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 supports read-while-write, has a small erase granularity, a short programming time and allows dual-bank booting.
It provides various security and protection mechanisms for code and data, read and write access.

Differences between STM32G43X/4X and STM32G47X/8X

	STM32G43X/4X (Category 2)	STM32G47X/8X (Category 3)	
		FLASH_OPTR[DBANK]=0 (Single bank)	FLASH_OPTR[DBANK]=1 (Dual bank)
Size	128 KB	512 KB	
Number of banks	1	1	2
Data width	64-bit	128-bit	64-bit
Page size	2-Kbyte	4-Kbyte	2-Kbyte
Flash organization	64 pages	128 pages	128 pages
Write Protect areas (WRPs)	2	4	2 per bank
Proprietary Code Read Protection Areas (PCROPs)	1	2	1 per bank
Securable memory area	1	2	1 per bank

life.augmented

This slide highlights the differences regarding the flash memory implementation between STM32G43X/4X, called category 2 microcontrollers, and STM32G47X/8X, called category 3 microcontrollers:
Flash memory size is 128 Kbytes for category 2, 512 Kbytes for category 3
Number of banks is 1 for category 2, 1 or 2 for category3, depending on the DBANK option bit. Note that read-while-write capability (or RWW) is only supported when the dual-bank architecture is active. This enables programming or erasing one bank while executing code from the other bank.
The page size which provides the minimum erase granularity is 2 KB for category 2, 4 KB for category 3 with single bank and 2 KB for category 3 with dual bank.
The number of pages is 64 for category 2 and 128 for category 3.

Regarding protection features, the category 2 microcontrollers have 1 Write Protect area, 1 PCROP and 1 securable memory area while category 3 microcontrollers have 2 write protect areas, 2 PCROPs and two securable memory areas.

Key features

4

- Page erase, bank erase and mass erase
- Fast erase (22 ms) and fast programming time (82 µs for double-words)
- 2 programming modes :
  - Standard (for main memory and OTP)
  - Fast (main memory only)
    - Programs 64 double-words without verifying the Flash locations
- 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

life.augmented

The Flash memory supports page erase, bank erase and mass erase.
A page, bank or mass erase operation requires only 22 ms, and the programming time is only 82 µs for a double-word.
Fast programming mode writes 64 double-words in a row and reduces the page programming time eliminating the need for verifying the Flash locations for each double-word access and in addition avoiding the rising and falling time of the high voltage for each double-word writing.
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®-M4.

Key features

5

- ART accelerator™ (Instruction cache, Data cache and prefetch buffer) enables a linear performance in relation to the frequency
- Protections:
  - Write Protection areas
  - Proprietary code read protection areas
  - WPR areas, 2 PCROP areas, 2 memory securable areas
  - Dual Bank mode: 2 WPR areas per bank, 1 PCROP area per bank,1 memory securable area per bank

life.augmented

The adaptive real-time memory accelerator, with an instruction cache, a data cache and a prefetch buffer, allows a linear performance in relation to the frequency. It also contributes to decrease the power consumption, as it belongs to the Vcore power domain.
The following protection mechanisms are supported.

- Write protection areas, used to protect against unwanted write operation.
- Proprietary code read protection areas (or PCROP): a part of the flash memory can be protected against access from third parties.
- The protected area is execute-only: it can only be reached by the STM32 CPU, as an instruction code area, while all other accesses (DMA, debug and CPU data read, write and erase) are strictly prohibited.
- The securable memory area defines an area of code which can be executed only once at boot, and never

again, unless a new reset occurs.

Flash memory organization (1/2)

6

The Flash memory is organized as follows:

- A Main memory block containing 64 or 128 pages
  - For Cat 3 devices with a single bank, Page size is 4-KB
    - Each page consists of 8 rows of 512 bytes
  - For Cat 3 microcontrollers with dual bank and Cat 2 devices, Page size is 2-KB
    - Each page consists of 8 rows of 256 bytes
- An Information block containing:
  - System memory which is reserved for use by ST and contains the **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
  - Option bytes for user configuration

life.augmented

The main memory contains 64 or 128 pages depending on the category of the microcontroller.
For Category 3 with a single-bank architecture, page size is 4 KB, each page consists of 8 rows of 512 bytes.
For Category 3 with dual-bank architecture and Category 2, page size is 2 KB, each page consists of 8 rows of 256 bytes.
In addition to main Flash memory, the STM32G4 supports:

- A System memory of 28 Kbytes containing the ST bootloader
- A 1-Kbyte OTP memory that can be used to store user data that must not be erased or modified. If one bit is ‘0’, the entire double-word can no longer be written, even with the value ‘0’.
- 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 (Cat 3, dual-bank)	Size	Name
Main memory	Bank 1	0x0800_0000– 0x0800_07FF	2 KB	Page 0
		…	…	…
		0x0803_F800– 0x0803_FFFF	2 KB	Page 127
	Bank 2	0x0804_0000– 0x0804_07FF	2 KB	Page 0
		…	…	…
		0x0807_F800– 0x0807_FFFF	2 KB	Page 127
Information block		0x1FFF 0000– 0x1FFF_6FFF	28 KB	System memory
		0x1FFF 7000– 0x1FFF_73FF	1 KB	OTP area
		0x1FFF 7800– 0x1FFF_787F	48 B	Option bytes

Operation	Granularity
Programming	8-Byte
Fast-programming	Row of 512 Bytes
Erase	Mass, bank and page
Securable memory	Page
Write protection	
Read protection	Global
Proprietary Code Readout Protection	Quadword (Cat 3 Single bank) or dword alignment

life.augmented

The first table details the memory organization based on a Main Flash memory area and an information block for Category 3 microcontrollers with dual-bank architecture. 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 512 bytes
- Erase is done either globally (mass erase) or with bank or page granularity.
- The securable memory is aligned on pages.
- Write protection is done per page
- Read protection is global
- Proprietary Code Readout Protection is based on programmable start and end addresses aligned on either quad-words or double-words.

Flash dual bank

Read-while write and Dual-bank boot capability

- Option DBANK in user option bytes selects dual bank mode
- Dual-bank Flash memory with dual-bank boot capability
  - Option BFB2 in user option bytes
    - BFB2 = 1, device boots either from Bank 2 or from Bank 1 depending on valid bank
    - BFB2 = 0, device boots in Bank 1 only
- Read-while-write
  - With its dual-bank capability, it is possible to read from one bank while programming/erasing the other bank
    - Code execution is not stopped when the Flash memory is being programmed
  - When programming/erasing data in the same bank: AHB is stalled as long as the program/erase operation is in progress

life.augmented

The DUALBANK (DBANK) option is used to select either a single bank or a dual bank for the category 3 devices. The Flash memory can be configured to support two banks, with read-while-write and dual-bank boot capability, able to boot from either Bank 1 or Bank 2. The BFB2 option in the user option bytes is used to select the dual-bank boot mode. When the BFB2 option is set, the device boots either from Bank 2 or Bank 1 depending on the valid bank. When the BFB2 option is cleared, the device always boots from Bank 1.

Flash read access

213 DMIPS at 170 MHz

- Adaptive Real-Time Accelerator (ART Accelerator™) enables a linear performance versus frequency, regardless of the Flash memory access time.

Wait states (WS) (FLASH latency)	HCLK (MHz)	
	V_{CORE} Range 1	V_{CORE} Range 2
0 WS	≤ 20	≤ 8
1 WS	≤ 40	≤ 16
2 WS	≤ 60	≤ 26
3 WS	≤ 80	-
4 WS	≤ 100	-
5 WS	≤ 120	-
6 WS	≤ 140	-
7 WS	≤ 160	-
8 WS	≤ 170	-

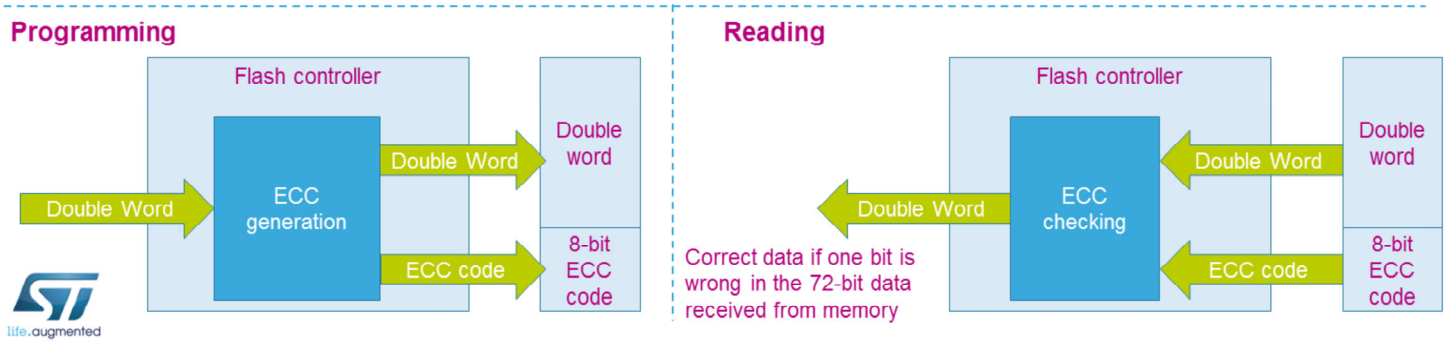
life.augmented

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 170 MHz with 7 wait states. It can be accessed with 0 wait states up to 20 MHz. For Range 2, it is up to 26 MHz, with 2 wait states. Thanks to the adaptive real-time 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 the frequency with a benchmark result of 213 Dhrystone MIPS at 170 MHz.

Flash memory features (1/2)

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**



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 including 8-bit ECC)
  - 144 bits when Cat 3 devices with single bank are used (72 bits x2)
- 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	82µs
One row (256 bytes) programming time	Standard mode: 2.61ms Fast mode: 1.91ms
One page (2 Kbytes) programming time	Standard mode: 20.91ms Fast mode: 15.29ms
Bank programming time	Standard mode: 2.68s Fast mode:1.96s
Page (2 Kbytes) erase time	22.02 ms
Mass erase time	22.13 ms

life.augmented

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

Fast programming is one third faster than standard mode programming.
Mass erase time, meaning a 512-Kbyte erase operation, approximately takes the same time as a page erase.

Row (64 double-word) Fast programming

13

- Only the **main memory** can be programmed with Fast programming
  - Neither 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
  - **Maximum** time between two double-word write requests is the programming time (approx. 50 µs) => Interrupts should be disabled
- The Flash memory clock frequency (HCLK) must be at least **8 MHz**

life.augmented

Fast programming vs standard programming:

- 512 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 50 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	2.61 ms	1.91 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

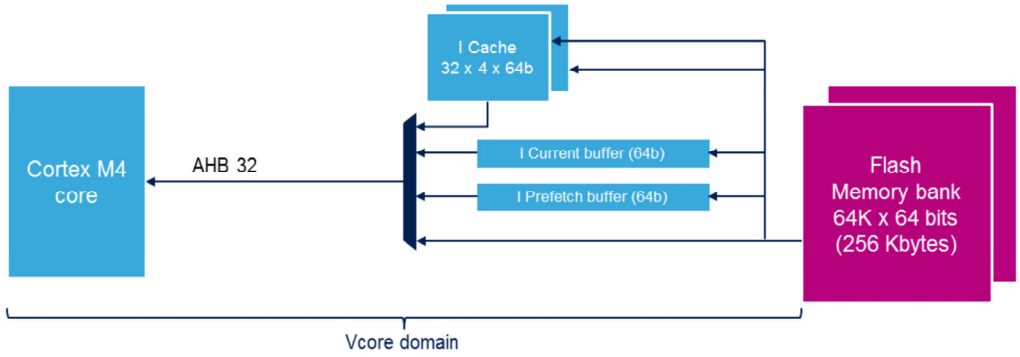
Each program / erase operation can degrade the Flash memory cell.
After an accumulation of program / erase cycles, memory cells can become non-functional, causing memory errors.
Endurance is the maximum number of erase/programming sequences that the Flash memory can support without affecting its reliability.
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.

Adaptive real-time memory accelerator (ART Accelerator™)

16

Outstanding performances and low-power

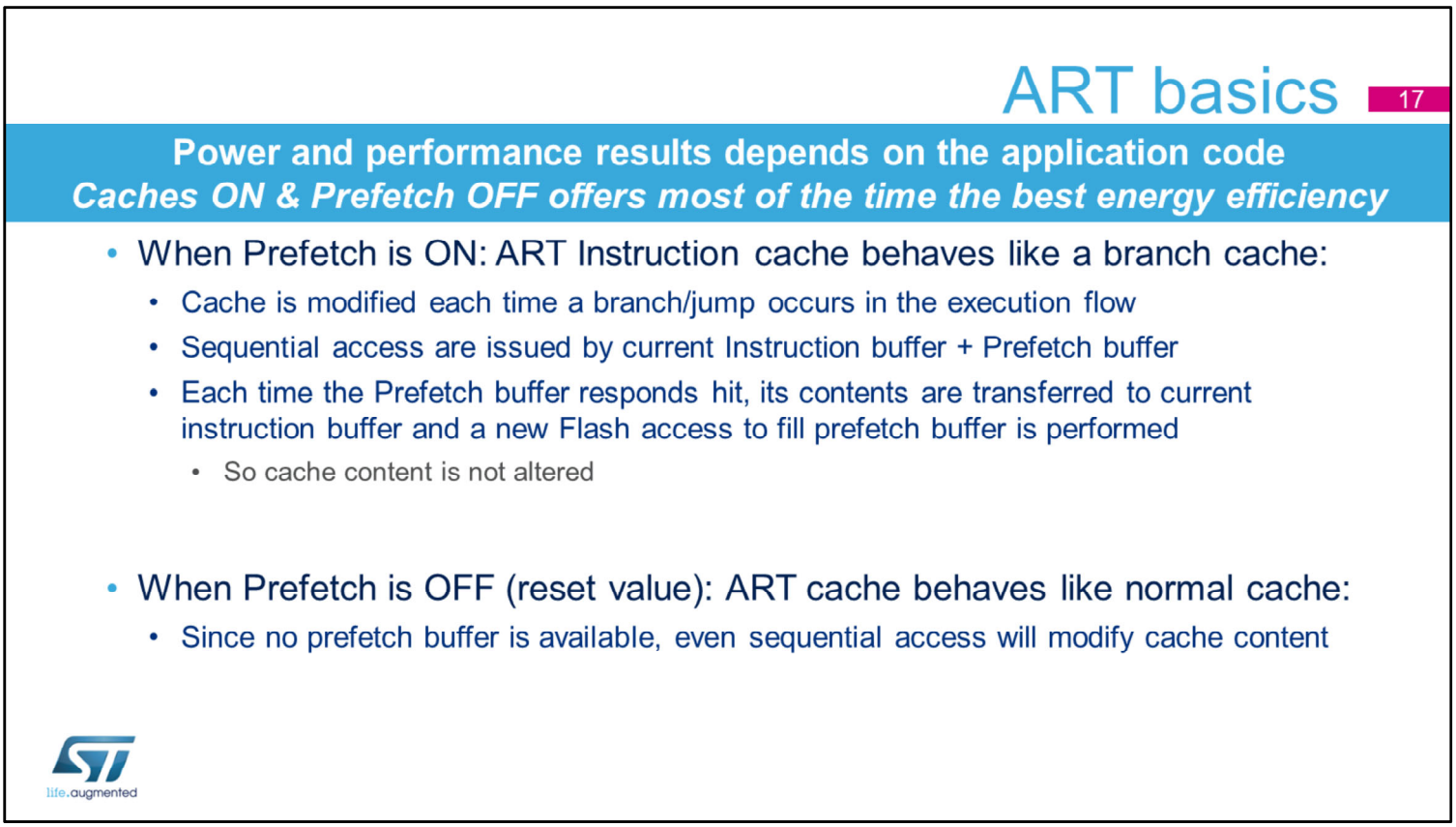
- **Instruction cache:** 32 lines of 4 double-words x 64 bits (1 Kbyte), for instructions
- **Data cache:** 8 lines of 4 double-words x 64 bits (256 bytes), for literal pools
- **Prefetch buffer:** one line of 64 bits
- Best tradeoff between cache size, power and performance



life.augmented

The ART accelerator brings outstanding performance and reduces dynamic power consumption.
It consists of a 1-KByte instruction cache, 256 bytes of data cache and a prefetch buffer.
The 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

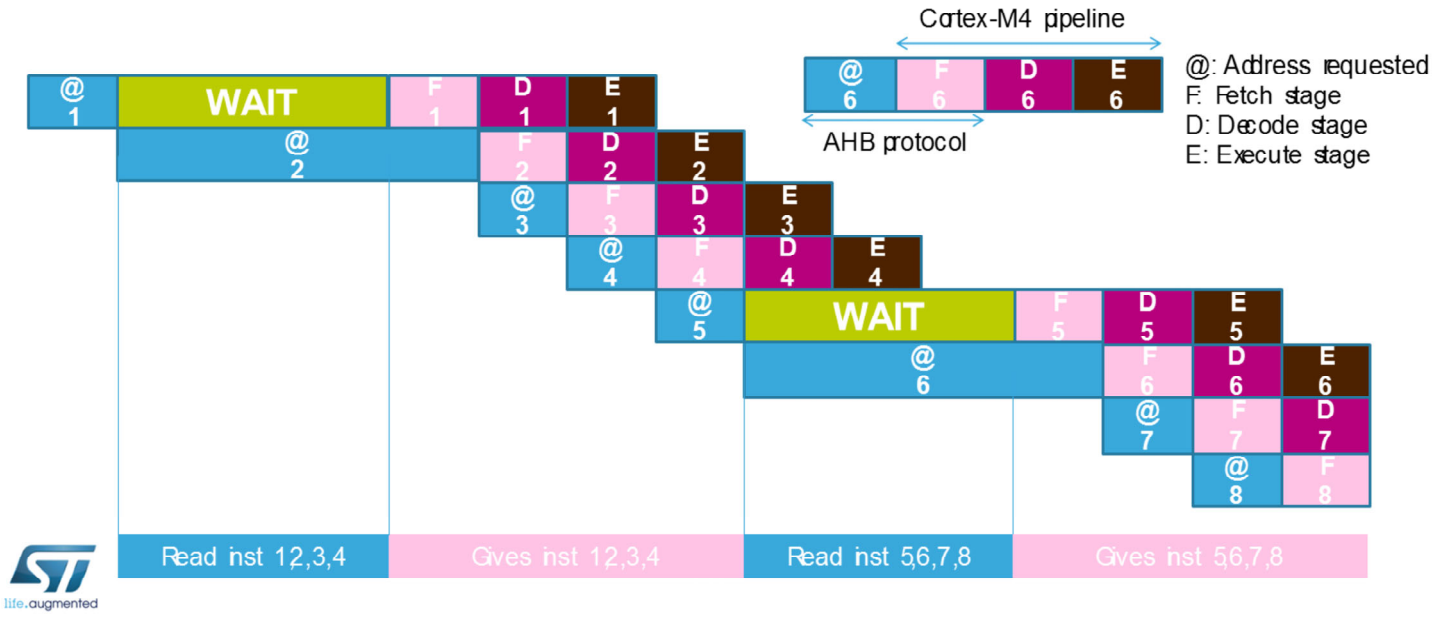
for power-saving. In a line, the 4 double-words may not all be valid.
In case of a miss, the Cortex M4 code takes the instruction directly from the Flash memory.
In parallel, the 64-bit line is copied into the current buffer enabled and I-Cache if enabled.
So the next sequential access is taken directly from the current buffer.
If prefetch is enabled, another 64-bit Flash access is performed to fill the Prefetch 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 Prefetch 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 Prefetch 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.



The Instruction Cache behaves differently depending on if the prefetch buffer is enabled or not.
If the prefetch 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 prefetch buffer; each time the prefetch buffer responds hit, its contents are transferred to the current instruction buffer and a new Flash access to fill the prefetch buffer is performed. In this case, the cache content is not altered.
If the prefetch buffer is disabled, the ART instruction cache behaves like a normal cache.
Since no prefetch buffer is available, even a sequential access will modify the cache content.
The power and performance tradeoff must be evaluated

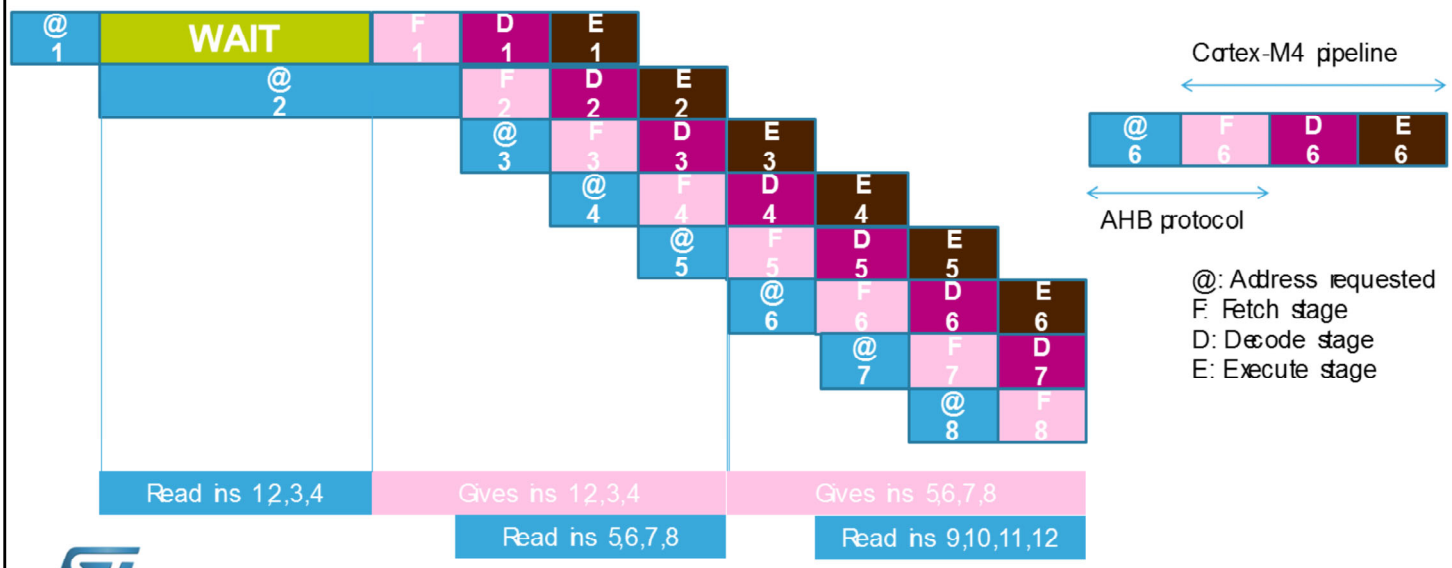
for each application to know whether is it is better to enable or disable the prefetch buffer.
For most of applications, enabling the prefetch buffer allows to increase slightly the performance but with a higher consumption.
Generally, the best energy efficiency is provided with caches enabled and prefetch buffer disabled, as it often reduces the number of Flash accesses.

Sequential 16-bit instructions execution (3WS), without prefetch



This slide shows the number of cycles needed to execute sequential 16-bit instructions without prefetch, when 3 wait states are needed to access the Flash memory. Every Flash access provides 64 bits or 4 instructions; 3 wait states are therefore inserted every 4 instructions, at every Flash access.

Sequential 16-bit instructions execution (3WS), with prefetch



This slide shows the number of cycles needed to execute sequential 16-bit instructions with prefetch enabled, when 3 wait states are needed to access the Flash memory.
After each Flash access, another Flash access is performed to fill the prefetch buffer.
So after all instructions are fetched from the current buffer, the next sequential instruction is read from the prefetch buffer and no wait state is inserted as long as the instruction flow is sequential.

Flash memory protection (1/2)

Flexible Flash memory protections according to application needs

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

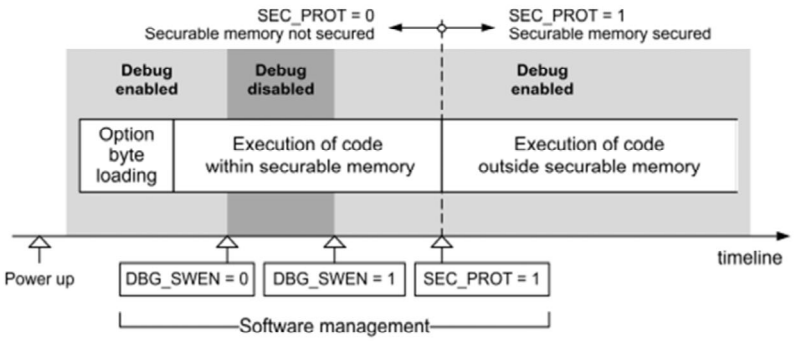
life.augmented

Several Flash memory protection options can be configured using the option bytes.
Readout protection aims to protect the contents of the Flash memory, option bytes, internal CCM 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®-M4 core.
PCROP areas are useful to protect only a part of the Flash memory against third party reads.
Write protection prevents part of the Flash memory from being erased and reprogrammed.

Flash memory protection (2/2)

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



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 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	Comment
BOR_LEV[2:0]	Brown-out reset threshold level	New in STM32G4
nRST_STOP; nRST_STDBY; nRST_SHDW	Reset / No reset generated when entering STOP/STANDBY/SHUTDOWN mode	Same as STM32F3
WWDG_SW; IDWG_SW IWDG_STOP; IWDG_STDBY	HW/SW window watchdog / independent watchdog Independent watchdog counter is frozen / not frozen in STOP/STANDBY mode	Same as STM32F3
BFB2	Dual-bank boot enable/disable	New in STM32G474
DBANK	Selection between single bank mode with 128-bit data read width and dual bank mode with 64 bits data read width	New in STM32G474
nBOOT1	Boot configuration (together with BOOT0 pin)	Same as STM32F3
nSWBOOT0	BOOT0 taken from the option bit nBOOT0 or taken from PB8/BOOT0 pin	New in STM32G4
nBOOT0		New is STM32G4
CCM SRAM_RST	CCM SRAM erase when system reset	New in STM32G4
SRAM_PE	SRAM1 and CCM SRAM parity check enable	Same as STM32F3

life.augmented

Option Bytes are used to early configure the system-on-chip before starting the Cortex®-M4. They represent 48 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 in the Option Bytes.

User option bytes

Options	Description	Comment
BOOT_LOCK	When set, it forces boot from main flash memory	New in STM32G4
SEC_SIZE1[7:0] SEC_SIZE2[7:0]	Bank 1 securable memory area size Bank 2 securable memory area size	New in STM32G4 No SEC_SIZE2 in STM32G431
IRHEN	Internal reset holder enable bit	New in STM32G474
NRST_MODE	PG10/NRST function selection	New in STM32G474

life.augmented

Bootlock forces the system to boot from the Main Flash memory regardless of the other boot options.

User option bytes (Security)

Options	Description	Comment
RDP[7:0]	Readout protection level	Same as F3
PCROP1_STRT[14:0] PCROP1_END[14:0] PCROP2_STRT[14:0] PCROP2_END[14:0]	Bank 1 PCROP area start offset address Bank 1 PCROP area end offset address Bank 2 PCROP area start offset address Bank 2 PCROP area end offset address	New in STM32G4
PCROP_RDP	PCROP area preserved when RDP level decreased	New in STM32G4
WRP1A_STRT[6:0] WRP1A_END[6:0] WRP1B_STRT[6:0] WRP1B_END[6:0] WRP2A_STRT[6:0] WRP2A_END[6:0] WRP2B_STRT[6:0] WRP2B_END[6:0]	Bank 1 Write protection area A start offset address Bank 1 Write protection area A end offset address Bank 1 Write protection area B start offset address Bank 1 Write protection area B end offset address Bank 2 Write protection area A start offset address Bank 2 Write protection area A end offset address Bank 2 Write protection area B start offset address Bank 2 Write protection area B end offset address	In STM32F3, the write protection is implemented with a granularity of 2 pages, with one option bit for every two pages

life.augmented

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.
- PCROPB_STRT and PCROPB_END define the proprietary code readout protection address range B.
- 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.

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 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 sequence error	Set by hardware when a write access to the Flash memory is performed by the code while PG or FSTPG have not been set previously. Set also by hardware when PROGERR, SIZERR, PGAERR, WRPERR, MISSERR or FASTERR is set due to a previous programming error.

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.

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	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.
Option validity error	Set by hardware when the options read may not be the one configured by the user.
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 is 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

- 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 can be configured in Power-down mode during Sleep/Low-power sleep modes
- Flash 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.
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 memory clock can be disabled if code is executed from SRAM and the Flash memory can be put in Power-down mode
Sleep	Active Flash memory clock can be disabled if code is executed from SRAM and the Flash memory can be put in Power-down 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 Flash memory clock can be disabled if code is executed from SRAM and the 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, Sleep, Low Power Run and Low Power Sleep modes, clock gating and power-down is supported. It can be used when code is executed from SRAM.

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.

Gating the clock and putting the Flash memory in Power-down mode significantly reduces power consumption.

Flash performance

3.36 Coremark / MHz

- Flash performance is almost linear with frequency thanks to the ART accelerator
- 3.36 CoreMark / MHz (Single Bank, Caches ON, Prefetch ON) => 571 CoreMark at 170MHz

	ART ON I-Cache ON D-Cache ON Prefetch ON		ART ON I-Cache ON D-Cache ON Prefetch OFF		ART OFF	
	Dual Bank	Single Bank	Dual Bank	Single Bank	Dual Bank	Single Bank
CoreMark / MHz	3.26	3.36	3.23	3.32	1.05	1.47

life.augmented

Here we compare code execution performance at 150 MHz while running the EEMBC CoreMark benchmark. The maximum performance is reached when the code is executed in CCM SRAM with data located in SRAM1. When executing from Flash memory at 150 MHz, the maximum CoreMark performance is reached when the ART accelerator is enabled, and there is almost no loss of performance due to the Flash access time requiring 7 wait states at 150 MHz. Enabling the prefetch buffer yields a slightly higher score: 3.36 CoreMark / MHz in case of Single bank mode.

Related peripherals

30

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

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)
- Memory protections.

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.