

## 4 Embedded flash memory (FLASH)

### 4.1 FLASH Introduction

The flash memory interface manages CPU (Cortex<sup>®</sup>-M0+) AHB accesses to the flash memory. It implements erase and program flash memory operations, read and write protection, and security mechanisms.

The flash memory interface accelerates code execution with a system of instruction prefetch and cache lines.

### 4.2 FLASH main features

- up to 256 Kbytes of flash memory (main memory)
  - up to 32 Kbytes for STM32C011xx and STM32C031xx
  - up to 64 Kbytes for STM32C051xx
  - up to 128 Kbytes for STM32C071xx
  - up to 256 Kbytes for STM32C091xx and STM32C092xx
- Memory organization:
  - One bank
  - Page size: 2 Kbytes
  - Subpage size: 512 bytes
- 64-bit wide data read (no ECC)
- Page erase (2 Kbytes) and mass erase

Flash memory interface features:

- Flash memory read operations
- Flash memory program/erase operations
- Read protection activated by option (RDP)
- Two write protection areas, selected by option (WRP)
- Two proprietary code read protection areas, selected by option (PCROP)
- Securable memory area
- Flash memory empty check
- Prefetch buffer
- CPU instruction cache: two cache lines of 64 bits (16-byte RAM)
- Option byte loader

### 4.3 FLASH functional description

#### 4.3.1 Flash memory organization

The flash memory is organized as 64-bit-wide memory cells that can be used for storing both code and data constants.

The flash memory is organized as follows:

- Main flash memory block containing up to 128 pages of 2 Kbytes, each page with 8 rows of 256 bytes
- Information block containing:
  - System memory from which the CPU boots in system memory boot mode. The area is reserved and contains the boot loader used to reprogram the flash memory through one of the interfaces listed in [Section 3.1.2: Embedded boot loader](#). On the manufacturing line, the devices are programmed and protected against spurious write/erase operations. For further details, refer to the AN2606 available from [www.st.com](http://www.st.com).
  - 1 Kbyte (128 double words) OTP (one-time programmable) for user data. The OTP data cannot be erased and can be written only once. If only one bit is at 0, the entire double word (64 bits) cannot be written anymore, even with the value 0x0000 0000 0000 0000.
  - Option bytes for user configuration.

The following table shows the mapping of the flash memory into information block and main memory area.

**Table 10. Flash memory organization for STM32C011xx and STM32C031xx**

Area	Addresses	Size (bytes)	Memory type
Information block	0x1FFF 7800 - 0x1FFF 7FFF	2 K (only first 128 bytes used)	Option bytes
	0x1FFF 7500 - 0x1FFF 77FF	768	Engineering bytes
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP
	0x1FFF 0000 - 0x1FFF 17FF	6 K	System memory
Main flash memory	0x0800 7800 - 0x0800 7FFF	2 K	Page 15
	...	...	...
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 0000 - 0x0800 07FF	2 K	Page 0

**Table 11. Flash memory organization for STM32C051xx**

Area	Addresses	Size (bytes)	Memory type
Information block	0x1FFF 7800 - 0x1FFF 7FFF	2 K	Option bytes
	0x1FFF 7500 - 0x1FFF 77FF	768	Engineering bytes
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP
	0x1FFF 0000 - 0x1FFF 2FFF	12 K	System memory

Table 11. Flash memory organization for STM32C051xx (continued)

Area	Addresses	Size (bytes)	Memory type
Main flash memory	0x0800 F800 - 0x0800 FFFF	2 K	Page 31
	...	...	...
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 0000 - 0x0800 07FF	2 K	Page 0

Table 12. Flash memory organization for STM32C071xx

Area	Addresses	Size (bytes)	Memory type
Information block	0x1FFF 7800 - 0x1FFF 7FFF	2 K	Option bytes
	0x1FFF 7500 - 0x1FFF 77FF	768	Engineering bytes
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP
	0x1FFF 0000 - 0x1FFF 6FFF	28 K	System memory
Main flash memory	0x0801 F800 - 0x0801 FFFF	2 K	Page 63
	...	...	...
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 0000 - 0x0800 07FF	2 K	Page 0

Table 13. Flash memory organization for STM32C091xx/92xx

Area	Addresses	Size (bytes)	Memory type
Information block	0x1FFF 7800 - 0x1FFF 7FFF	2 K	Option bytes
	0x1FFF 7500 - 0x1FFF 77FF	768	Engineering bytes
	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP
	0x1FFF 0000 - 0x1FFF 3FFF	16 K	System memory
Main flash memory	0x0803 F800 - 0x0803 FFFF	2 K	Page 127
	...	...	...
	0x0800 1000 - 0x0800 17FF	2 K	Page 2
	0x0800 0800 - 0x0800 0FFF	2 K	Page 1
	0x0800 0000 - 0x0800 07FF	2 K	Page 0

### 4.3.2 FLASH read access latency

To correctly read data from the flash memory, set the LATENCY[2:0] bitfield of the *FLASH access control register (FLASH\_ACR)* register as per the following table.

**Table 14. LATENCY[2:0] setting as function of HCLK frequency**

HCLK (MHz)	LATENCY[2:0]
≤ 24	000 (1 HCLK cycle)
≤ 48	001 (2 HCLK cycles)

Upon power-on reset or upon wake-up from Standby, the HCLK clock frequency is automatically set to 12 MHz and the LATENCY[2:0] bitfield to 000. See [Section 6.2: Clocks](#).

To change HCLK frequency, respect the following sequence:

#### Increasing HCLK frequency

1. Set the LATENCY[2:0] bitfield to correspond to the target HCLK frequency, as per [Table 14](#).
2. Read the LATENCY[2:0] bitfield until it returns the value written in the previous step.
3. Select the system clock source as required, through the SW[2:0] bitfield of the RCC\_CFGR register.
4. Set the HCLK clock prescaler as required, through the HPRE[3:0] bitfield of the RCC\_CFGR register.

The clock source effectively selected for the system can be checked by reading the clock source status bitfield SWS[2:0] of the RCC\_CFGR register. The HPRE[3:0] bitfield of the RCC\_CFGR register can also be read to check its content.

#### Decreasing HCLK frequency

1. Select the system clock source as required, through the SW[2:0] bitfield of the RCC\_CFGR register.
2. Set the HCLK clock prescaler as required, through the HPRE[3:0] bitfield of the RCC\_CFGR register.
3. Read the SWS[2:0] bitfield of the RCC\_CFGR register until it returns the value set into SW[2:0] in step 1. The HPRE[3:0] bitfield of the RCC\_CFGR register can also be read to check its content.
4. Set the LATENCY[2:0] bitfield to correspond to the target HCLK frequency, as per [Table 14](#).

### 4.3.3 Flash memory acceleration

#### Instruction prefetch

Each flash memory read operation provides 64 bits from either two instructions of 32 bits or four instructions of 16 bits according to the program launched. This 64-bits current instruction line is saved in a current buffer. So, in case of sequential code, at least two CPU cycles are needed to execute the previous read instruction line. Prefetch on the CPU S-bus can be used to read the next sequential instruction line from the flash memory while the current instruction line is being requested by the CPU.

Prefetch is enabled by setting the PRFTEN bit of the *FLASH access control register (FLASH\_ACR)*. This feature is useful if at least one wait state is needed to access the flash memory.

When the code is not sequential (branch), the instruction may not be present in the currently used instruction line or in the prefetched instruction line. In this case (miss), the penalty in terms of number of cycles is at least equal to the number of wait states.

If a loop is present in the current buffer, no new access is performed.

### Cache memory

To limit the time lost due to jumps, it is possible to retain two cache lines of 64 bits (16 bytes) in the instruction cache memory. This feature can be enabled by setting the instruction cache enable (ICEN) bit of the *FLASH access control register (FLASH\_ACR)*. Each time a miss occurs (requested data not present in the currently used instruction line, in the prefetched instruction line or in the instruction cache memory), the line read is copied into the instruction cache memory. If some data contained in the instruction cache memory are requested by the CPU, they are 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.

The Instruction cache memory is enabled after system reset.

No data cache is available on Cortex<sup>®</sup>-M0+.

## 4.3.4 FLASH program and erase operations

The device-embedded flash memory can be programmed using in-circuit programming or in-application programming.

The **in-circuit programming (ICP)** method is used to update the entire contents of the flash memory, using SWD protocol or the supported interfaces by the system boot loader, to load the user application for the CPU, into the microcontroller. ICP offers quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

In contrast to the ICP method, **in-application programming (IAP)** can use any communication interface supported by the microcontroller (I/Os, UART, I<sup>2</sup>C, SPI, etc.) to download programming data into memory. IAP allows the user to re-program the flash memory while the application is running. Nevertheless, part of the application has to have been previously programmed in the flash memory using ICP.

The success of a data word programming operation and a page/bank erase operation is not guaranteed if aborted due to device reset or power loss.

During a program/erase operation to the flash memory, any attempt to read the flash memory stalls the bus. The read operation proceeds correctly once the program/erase operation has completed.

### Unlocking the flash memory

After reset, write into the *FLASH control register (FLASH\_CR)* is not allowed so as to protect the flash memory against possible unwanted operations due, for example, to electric disturbances. The following sequence unlocks these registers:

1. Write KEY1 = 0x4567 0123 in the *FLASH key register (FLASH\_KEYR)*
2. Write KEY2 = 0xCDEF 89AB in the *FLASH key register (FLASH\_KEYR)*.

Any wrong sequence locks the FLASH\_CR registers until the next system reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

The FLASH\_CR registers can be locked again by software by setting the LOCK bit in one of these registers.

**Note:** *The FLASH\_CR register cannot be written when the BSY1 bit of the [FLASH status register \(FLASH\\_SR\)](#) is set. Any attempt to write to this register with the BSY1 bit set causes the AHB bus to stall until the BSY1 bit is cleared.*

### 4.3.5 FLASH main memory erase sequences

The flash memory erase operation can be performed at page level (page erase), or on the whole memory (mass erase). Mass erase does not affect the information block (system flash memory, OTP and option bytes).

#### Flash memory page erase

When a page is protected by PCROP or WRP, it is not erased and the WRPERR bit is set.

**Table 15. Page erase overview**

SEC_PROT	PCROP	WRP	PCROP_RDP	Comment	WRPERR	CPU bus error
0	No	No	x	Page is erased	No	No
	No	Yes		Page erase aborted (no page erase started)	Yes	
	Yes	No				
	Yes	Yes				
1	x			Protected pages only	No	Yes

To erase a page (2 Kbytes), follow the procedure below:

1. Check that no flash memory operation is ongoing by checking the BSY1 bit of the [FLASH status register \(FLASH\\_SR\)](#).
2. Check and clear all error programming flags due to a previous programming. If not, PGSERR is set.
3. Check that the CFGBSY bit of the [FLASH status register \(FLASH\\_SR\)](#) is cleared.
4. Set the PER bit and select the page to erase (PNB) in the [FLASH control register \(FLASH\\_CR\)](#).
5. Set the STRT bit of the [FLASH control register \(FLASH\\_CR\)](#).
6. Wait until the CFGBSY bit of the [FLASH status register \(FLASH\\_SR\)](#) is cleared.

**Note:** *The HSI48 internal oscillator (with a divide by three to provide 16 MHz) is automatically enabled when the STRT bit is set. It is automatically disabled when the STRT bit is cleared, except if previously enabled with the HSION bit of the RCC\_CR register.*

#### Flash memory bank or mass erase

When PCROP or WRP is enabled, the flash memory mass erase is aborted, no erase starts, and the WRPERR bit is set.

Table 16. Mass erase overview

SEC_PROT	PCROP	WRP	PCROP_RDP	Comment	WRPERR	CPU bus error
0	No	No	x	Memory is erased	No	No
	No	Yes		Erase aborted (no erase started)	Yes	
	Yes	No				
	Yes	Yes				
1	x			Erase aborted (no erase started)	No	Yes

To perform a mass erase, respect the following procedure:

1. Check that no flash memory operation is ongoing by checking the BSY1 bit of the [FLASH status register \(FLASH\\_SR\)](#).
2. Check and clear all error programming flags due to a previous programming. If not, PGSERR is set.
3. Check that the CFGBSY bit of the [FLASH status register \(FLASH\\_SR\)](#) is cleared.
4. Set the MER1 bit of the [FLASH control register \(FLASH\\_CR\)](#).
5. Set the STRT bit of the [FLASH control register \(FLASH\\_CR\)](#).
6. Wait until the CFGBSY bit of the [FLASH status register \(FLASH\\_SR\)](#) is cleared.

**Note:** The HSI48 internal oscillator (with a divide by three to provide 16 MHz) is automatically enabled when the STRT bit is set. It is automatically disabled when the STRT bit is cleared, except if previously enabled with the HSION bit of the RCC\_CR register.

#### 4.3.6 FLASH main memory programming sequences

The flash memory is programmed 64 bits at a time.

Programming a previously programmed address with a non-zero data is not allowed. Any such attempt sets PROGERR flag of the [FLASH status register \(FLASH\\_SR\)](#).

It is only possible to program a double word (2 x 32-bit data).

- Any attempt to write byte (8 bits) or half-word (16 bits) sets SIZERR flag of the [FLASH status register \(FLASH\\_SR\)](#).
- Any attempt to write a double word that is not aligned with a double word address sets PGAERR flag of the [FLASH status register \(FLASH\\_SR\)](#).

##### Standard programming

The flash memory programming sequence in standard mode is as follows:

1. Check that no main flash memory operation is ongoing by checking the BSY1 bit of the *FLASH status register (FLASH\_SR)*.
2. Check and clear all error programming flags due to a previous programming. If not, PGSERR is set.
3. Check that the CFGBSY bit of the *FLASH status register (FLASH\_SR)* is cleared.
4. Set the PG bit of the *FLASH control register (FLASH\_CR)*.
5. Perform the data write operation at the desired memory address, inside main flash memory block or OTP area. Only double word (64 bits) can be programmed.
  - a) Write a first word in an address aligned with double word
  - b) Write the second word.
6. Wait until the CFGBSY bit of the *FLASH status register (FLASH\_SR)* is cleared.
7. Check that the EOP flag in the *FLASH status register (FLASH\_SR)* is set (programming operation succeeded), and clear it by software.
8. Clear the PG bit of the *FLASH control register (FLASH\_CR)* if there no more programming request anymore.

**Note:** *When the flash memory interface has received a good sequence (a double word), programming is automatically launched and the BSY1 bit set. The HSI48 internal oscillator (with a divide by three to provide 16 MHz) is automatically enabled when the PG bit is set. It is automatically disabled when the PG bit is cleared, except if previously enabled with the HSION bit of the RCC\_CR register.*

### Fast programming

The main purpose of this mode 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.

This mode allows programming a row (32 double words = 256 bytes).

During fast programming, the flash memory clock (HCLK) frequency must be at least 8 MHz.

Only the main flash memory can be programmed in Fast programming mode.

The main flash memory programming sequence in fast mode is described below:

1. Perform a mass or page erase. If not, PGSERR is set.
2. Check that no main flash memory operation is ongoing by checking the BSY1 bit of the *FLASH status register (FLASH\_SR)*.
3. Check and clear all error programming flag due to a previous programming.
4. Check that the CFGBSY bit of the *FLASH status register (FLASH\_SR)* is cleared.
5. Set the FSTPG bit in *FLASH control register (FLASH\_CR)*.
6. Write 32 double-words to program a row (256 bytes).
7. Wait until the CFGBSY bit of the *FLASH status register (FLASH\_SR)* is cleared.
8. Check that the EOP flag in the *FLASH status register (FLASH\_SR)* is set (programming operation succeeded), and clear it by software.
9. Clear the FSTPG bit of the *FLASH status register (FLASH\_SR)* if there are no more programming requests anymore.



**Note:** *When attempting to write in Fast programming mode while a read operation is ongoing, the programming is aborted without any system notification (no error flag is set).*

*When the flash memory interface has received the first double word, programming is automatically launched. The BSY1 bit is set when the high voltage is applied for the first double word, and it is cleared when the last double word has been programmed or in case of error.*

*The HSI48 internal oscillator (with a divide by three to provide 16 MHz) is automatically enabled when the FSTPG bit is set. It is automatically disabled when the FSTPG bit is cleared, except if previously enabled with the HSION bit of the RCC\_CR register.*

*The 32 double words must be written successively. The high voltage is kept on the flash memory for all the programming. Maximum time between two double words write requests is the time programming (around 20  $\mu$ s). If a second double word arrives after this time programming, fast programming is interrupted and MISSERR is set.*

*High voltage must not exceed 8 ms for a full row between two erases. This is guaranteed by the sequence of 32 double words successively written with a clock system greater or equal to 8 MHz. An internal time-out counter counts 7 ms when Fast programming is set and stops the programming when time-out is over. In this case the FASTERR bit is set.*

*If an error occurs, high voltage is stopped and next double word to programmed is not programmed. Anyway, all previous double words have been properly programmed.*

### Programming errors

Several kind of errors can be detected. In case of error, the flash memory operation (programming or erasing) is aborted.

- **PROGERR:** Programming error  
In standard programming: PROGERR is set if the word to write is not previously erased (except if the value to program is full zero and the target address is in the main flash memory).
- **SIZERR:** Size programming error  
In standard programming or in fast programming: only double word can be programmed, and only 32-bit data can be written. SIZERR is set if a byte or an half-word is written.
- **PGAERR:** Alignment programming error  
PGAERR is set if one of the following conditions occurs:
  - In standard programming: the first word to be programmed is not aligned with a double word address, or the second word doesn't belong to the same double word address.
  - In fast programming: the data to program doesn't belong to the same row than the previous programmed double words, or the address to program is not greater than the previous one.
- **PGSERR:** Programming sequence error  
PGSERR is set if one of the following conditions occurs:
  - In the standard programming sequence or the fast programming sequence: a data is written when PG and FSTPG are cleared.
  - In the standard programming sequence or the fast programming sequence: MER1 and PER are not cleared when PG or FSTPG is set.
  - In the fast programming sequence: the Mass erase is not performed before setting the FSTPG bit.

- In the mass erase sequence: PG, FSTPG, and PER are not cleared when MER1 is set.
- In the page erase sequence: PG, FSTPG and MER1 are not cleared when PER is set.
- PGSERR is set also if PROGERR, SIZERR, PGAERR, WRPERR, MISSERR, FASTERR or PGSERR is set due to a previous programming error.
- **WRPERR:** Write protection error  
WRPERR is set if one of the following conditions occurs:
  - Attempt to program or erase in a write protected area (WRP) or in a PCROP area.
  - Attempt to perform a mass erase when one page or more is protected by WRP or PCROP.
  - The debug features are connected or the boot is executed from SRAM or from system flash memory when the read protection (RDP) is set to level 1.
  - Attempt to modify the option bytes when the read protection (RDP) is set to level 2.
- **MISSERR:** Fast programming data miss error  
In fast programming: all the data must be written successively. MISSERR is set if the previous data programming is finished and the next data to program is not written yet.
- **FASTERR:** Fast programming error  
In fast programming: FASTERR is set if one of the following conditions occurs:
  - when FSTPG bit is set for more than 8 ms, which generates a time-out detection
  - when the row fast programming has been interrupted by a MISSERR, PGAERR, WRPERR or SIZERR

If an error occurs during a program or erase operation, one of the following error flags of the [FLASH status register \(FLASH\\_SR\)](#) is set:

- PROGERR, SIZERR, PGAERR, PGSERR, MISSERR (program error flags)
- WRPERR (protection error flag)

In this case, if the error interrupt enable bit ERRIE of the [FLASH control register \(FLASH\\_CR\)](#) is set, an interrupt is generated and the operation error flag OPERR of the [FLASH status register \(FLASH\\_SR\)](#) is set.

*Note:* If several successive errors are detected (for example, in case of DMA transfer to the flash memory), the error flags cannot be cleared until the end of the successive write request.

### Programming and cache

If an erase operation in flash memory also concerns data in the instruction cache, the user has to ensure that these data are rewritten before they are accessed during code execution.

*Note:* The cache should be flushed only when it is disabled (ICEN = 0).

## 4.4 FLASH option bytes

### 4.4.1 FLASH option byte description

The option bytes are configured by the end user depending on the application requirements. As a configuration example, the watchdog may be selected in hardware or software mode (refer to [Section 4.4.2: FLASH option byte programming](#)).

A double word is split up in option bytes as indicated in [Table 17](#).

**Table 17. Option byte format**

63-56	55-48	47-40	39-32	31-24	23-16	15 -8	7-0
Complemented option byte 3	Complemented option byte 2	Complemented option byte 1	Complemented option byte 0	Option byte 3	Option byte 2	Option byte 1	Option byte 0

[Table 18](#) shows the organization of the option bytes (the lower word only) in the flash memory information block. The software can read the option bytes from these flash memory locations or from their corresponding option registers referenced in the table. Refer to sections [4.7.6](#) to [4.7.13](#) for the description of the option register bitfields, also applicable to the option byte bitfields.

**Table 18. Organization of option bytes**

Address <sup>(1)</sup>	Corresponding option register (section)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		Option byte 3								Option byte 2								Option byte 1								Option byte 0								
0x1FFF7800	FLASH_OPTR (4.7.6)	FDCAN_BL_CK[1:0]  IRHEN  NRST_MODE[1:0]  NBOOT0 NBOOT1  NBOOT_SEL SECURE_MUXING_EN RAM_PARITY_CHECK HSE_NOT_REMAPPED <sup>(2)</sup> Reserved WWDG_SW IWDG_STBY IWDG_STOP IWDG_SW NRST_SHDW NRST_STDBY NRST_STOP BORF_LEV BORR_LEV BOR_EN								RDP																								
	Factory value	0	0	1	1	1	1	1	1	1	1	1	X	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	1	0	1	0
0x1FFF7808	FLASH_PCROP1ASR (4.7.7)	Reserved																								PCROP1A_STRT								
	Factory value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1	1
0x1FFF7810	FLASH_PCROP1AER (4.7.8)	PCROP_RDP	Reserved																								PCROP1A_END							
	Factory value		0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0	0	
0x1FFF7818	FLASH_WRP1AR (4.7.9)	Reserved								WRP1A_END								Reserved								WRP1A_STRT								
	Factory value	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1
0x1FFF7820	FLASH_WRP1BR (4.7.10)	Reserved								WRP1B_END								Reserved								WRP1B_STRT								
	Factory value	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1
0x1FFF7828	FLASH_PCROP1BSR (4.7.11)	Reserved																								PCROP1B_STRT								
	Factory value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	1	1	1	1	1	1	1	1	

Table 18. Organization of option bytes (continued)

Address <sup>(1)</sup>	Corresponding option register (section)	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
		Option byte 3								Option byte 2								Option byte 1								Option byte 0								
0x1FFF7830	FLASH_PCROP1BER (4.7.12)	Reserved																								PCROP1B_END								
	Factory value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	0		
0x1FFF7870	FLASH_SECR (4.7.13)	Reserved																BOOT_LOCK	Reserved								SEC_SIZE							
	Factory value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	0		X	X	X	X	X	X	X	X	0	0	0	0	0	0	0	

1. The upper 32 bits of the double-word address contain the inverted data from the lower 32 bits.
2. Only relevant to products in packages with 48 to 64 pins.

#### 4.4.2 FLASH option byte programming

After reset, the option related bits of the *FLASH control register (FLASH\_CR)* are write-protected. To run any operation on the option byte page, the option lock bit OPTLOCK of the *FLASH control register (FLASH\_CR)* must be cleared. The following sequence is used to unlock this register:

1. Unlock the FLASH\_CR with the LOCK clearing sequence (refer to *Unlocking the flash memory*)
2. Write OPTKEY1 = 0x0819 2A3B of the *FLASH option key register (FLASH\_OPTKEYR)*
3. Write OPTKEY2 = 0x4C5D 6E7F of the *FLASH option key register (FLASH\_OPTKEYR)*

Any wrong sequence locks up the flash memory option registers until the next system reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

The user options can be protected against unwanted erase/program operations by setting the OPTLOCK bit by software.

**Note:** If LOCK is set by software, OPTLOCK is automatically set as well.

#### Modifying user options

The option bytes are programmed differently from a main flash memory user address.

To modify the value of user options, follow the procedure below:

1. Clear OPTLOCK option lock bit with the clearing sequence described above
2. Write the desired values in the FLASH option registers.
3. Check that no flash memory operation is ongoing, by checking the BSY1 bit of the *FLASH status register (FLASH\_SR)*.
4. Set the Options Start bit OPTSTRT of the *FLASH control register (FLASH\_CR)*.
5. Wait for the BSY1 bit to be cleared.

**Note:** Any modification of the value of one option is automatically performed by erasing user option byte pages first, and then programming all the option bytes with the values contained in the flash memory option registers.

The complementary values are automatically computed and written into the complemented option bytes upon setting the OPTSTRT bit.

**Caution:** Upon an option byte programming failure (for any reason, such as loss of power or a reset during the option byte change sequence), the mismatch values of the option bytes are loaded after reset. Those mismatch values force a secure configuration that might block the code execution. To prevent this, only program option bytes in a safe environment – safe supply, no pending watchdog, and clean reset line.

### Option byte loading

After the BSY1 bit is cleared, all new options are updated into the flash memory, but not applied to the system. A read from the option registers still returns the last loaded option byte values, the new options has effect on the system only after they are loaded.

Option byte loading is performed in two cases:

- when OBL\_LAUNCH bit of the *FLASH control register (FLASH\_CR)* is set
- after a power reset (BOR reset or exit from Standby/Shutdown modes)

Option byte loader performs a read of the options block and stores the data into internal option registers. These internal registers configure the system and can be read by software. Setting OBL\_LAUNCH generates a reset so the option byte loading is performed under system reset.

Each option bit has also its complement in the same double word. During option loading, a verification of the option bit and its complement allows to check the loading has correctly taken place.

During option byte loading, the options are read by double word.

If the word and its complement are matching, the option word/byte is copied into the option register.

If the comparison between the word and its complement fails, a status bit OPTVERR is set. Mismatch values are forced into the option registers:

- For USR OPT option, the value of mismatch is 1 for all option bits, except the BOR\_EN bit that is 0 (BOR disabled).
- For WRP option, the value of mismatch is the default value “No protection”.
- For RDP option, the value of mismatch is the default value “level 1”.
- For PCROP, the value of mismatch is “all memory protected”.
- For BOOT\_LOCK, the value of mismatch is “boot forced from main flash memory”.

**Note:** In this situation (mismatch of option bytes), setting both BOOT\_LOCK and RDP level 1 does not disable the debug capabilities, to allow the part reprogramming.

Upon system reset, the option bytes are copied into the following option registers that can be read and written by software:

- FLASH\_OPTR
- FLASH\_PCROP1xSR (x = A or B)
- FLASH\_PCROP1xER (x = A or B)
- FLASH\_WRP1xR (x = A or B)
- FLASH\_SECR

These registers are also used to modify options. If these registers are not modified by user, they reflect the options states of the system. See [Modifying user options](#) for more details.

## 4.5 Flash memory protection

The main flash memory can be protected against external accesses with the read protection (RDP). The pages can also be protected against unwanted write (WRP) due to loss of program counter context. The write-protection WRP granularity is 2 Kbytes. Apart from the RDP and WRP, the flash memory can also be protected against read and write by third party (PCROP). The PCROP granularity (subpage size) is 512 bytes.

### 4.5.1 FLASH read protection (RDP)

The read protection is activated by setting the RDP option byte and then, by applying a system reset to reload the new RDP option byte. The read protection protects the main flash memory, the option bytes.

There are three levels of read protection from no protection (level 0) to maximum protection or no debug (level 2).

The flash memory is protected when the RDP option byte and its complement contain the pair of values shown in [Table 19](#).

**Table 19. Flash memory read protection status**

RDP byte value	RDP complement byte value	Read protection level
0xAA	0x55	Level 0
Any values except the combinations [0xAA, 0x55] and [0xCC, 0x33]		Level 1 (default)
0xCC	0x33	Level 2

The system memory area is read-accessible whatever the protection level. It is never accessible for program/erase operation.

#### Level 0: no protection

Read, program and erase operations within the main flash memory area are possible. The option bytes are also accessible by all operations.

### Level 1: Read protection

Level 1 read protection is set when the RDP byte and the RDP complemented byte contain any value combinations other than [0xAA, 0x55] and [0xCC, 0x33]. Level 1 is the default protection level when RDP option byte is erased.

- **User mode:** Code executing in user mode (boot from user flash memory) can access main flash memory and option bytes with all operations.
- **Debug, boot from SRAM, and boot from system memory modes:** In debug mode or when code boots from SRAM or system memory, the main flash memory is totally inaccessible. In these modes, a read or write access to the flash memory generates a bus error and a Hard Fault interrupt.

**Caution:** In level 1 with a PCROP area defined, user code to protect by RDP but not by PCROP must be placed outside pages containing a PCROP-protected subpage.

### Level 2: No debug

In this level, the protection level 1 is guaranteed. In addition, the CPU debug port, the boot from RAM (boot RAM mode) and the boot from system memory (boot loader mode) are no more available. In user execution mode (boot FLASH mode), all operations are allowed on the main flash memory.

*Note:* The CPU debug port is also disabled under reset.

*Note:* STMicroelectronics is not able to perform analysis on defective parts on which the level 2 protection has been set.

### Changing the read protection level

The read protection level can change:

- from level 0 to level 1, upon changing the value of the RDP byte to any value except 0xCC
- from level 0 or level 1 to level 2, upon changing the value of the RDP byte to 0xCC
- from level 1 to level 0, upon changing the value of the RDP byte to 0xAA

Once in level 2, it is no more possible to modify the read protection level.

When the read protection is changed from level 0 during or after (since last power on) the debugger is connected or the MCU boot from system memory / SRAM, to reload the option byte, apply a POR (power-on reset) instead of a system reset / OBL\_LAUNCH. Otherwise, the internal read out protection is activated and any data read triggers a hard fault. If the read protection is programmed through the software, the POR can be done by a transition to Standby (or Shutdown) mode followed by a wake-up.

With the PCROP\_RDP bit of the [FLASH PCROP area A end address register \(FLASH\\_PCROP1AER\)](#) set, the change from level 1 to level 0 triggers full mass erase of the main flash memory. The user options except PCROP protection are set to their previous values copied from FLASH\_OPTR and FLASH\_WRP1xR (x = A or B). PCROP is disabled. The OTP area is not affected by mass erase and remains unchanged.

With the PCROP\_RDP bit cleared, a partial mass erase occurs, only erasing flash memory pages that do not overlap with PCROP area (do not contain any PCROP-protected subpage). The option bytes are re-programmed with their previous values. This is also true for FLASH\_PCROP1xSR and FLASH\_PCROP1xER registers (x = A or B).

Table 20: Mass erase upon RDP regression from level 1 to level 0

PCROP area	PCROP_RDP	Mass erase
None	x	Full
Part of flash memory	1	
	0	Partial (flash memory pages not overlapping with PCROP area)
Full flash memory		None

**Note:** Mass erase (full or partial) is only triggered by the RDP regression from level 1 to level 0. RDP level increase (level 0 to level 1, 1 to 2, or 0 to 2) does not cause any mass erase. To validate the protection level change, the option bytes must be reloaded by setting the OBL\_LAUNCH bit of the [FLASH control register \(FLASH\\_CR\)](#).

Figure 3. Changing read protection (RDP) level

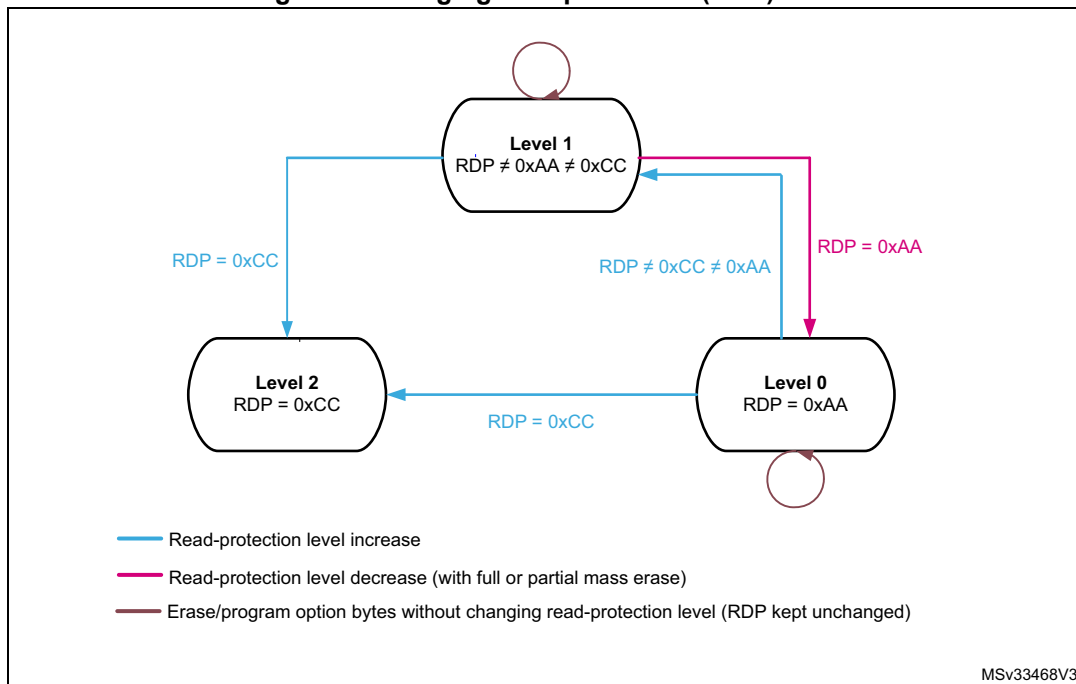


Table 21. Access status versus protection level and execution modes

Area	Protection level	User execution (BootFromFlash)			Debug/ BootFromRam/ BootFromLoader		
		Read	Write	Erase	Read	Write	Erase
Main flash memory	1	Yes	Yes	Yes	No	No	No <sup>(3)</sup>
	2	Yes	Yes	Yes	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>
System memory <sup>(2)</sup>	1	Yes	No	No	Yes	No	No
	2	Yes	No	No	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>



Table 21. Access status versus protection level and execution modes (continued)

Area	Protection level	User execution (BootFromFlash)			Debug/ BootFromRam/ BootFromLoader		
		Read	Write	Erase	Read	Write	Erase
Option bytes	1	Yes	Yes <sup>(3)</sup>	Yes	Yes	Yes <sup>(3)</sup>	Yes
	2	Yes	No	No	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>
OTP	1	Yes	Yes	N/A	Yes	No	N/A
	2	Yes	Yes	N/A	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>	N/A <sup>(1)</sup>

1. When the protection level 2 is active, the Debug port, the boot from RAM and the boot from system memory are disabled.
2. The system memory is only read-accessible, whatever the protection level (0, 1 or 2) and execution mode.
3. The main flash memory is erased when the RDP option byte is programmed with all level of protections disabled (0xAA).

#### 4.5.2 FLASH proprietary code readout protection (PCROP)

Two areas of the flash memory can be protected against unwanted read and/or write by a third party.

The protected area is execute-only: it can only be reached by the STM32 CPU, with an instruction code, while all other accesses (DMA, debug and CPU data read, write and erase) are strictly prohibited. The PCROP areas have subpage (512-byte) granularity. An additional option bit (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 (refer to [Changing the read protection level](#)).

Each PCROP area is defined by a start subpage offset and an end subpage offset into the flash memory. These offsets are defined with the corresponding bitfields of the PCROP address registers [FLASH PCROP area A start address register \(FLASH\\_PCROP1ASR\)](#), [FLASH PCROP area A end address register \(FLASH\\_PCROP1AER\)](#), [FLASH PCROP area B start address register \(FLASH\\_PCROP1BSR\)](#), and [FLASH PCROP area B end address register \(FLASH\\_PCROP1BER\)](#).

A PCROP area *x* (A or B) is defined from the address:

*flash memory base address* + [*PCROP1x\_STRT* x 0x200] (included)

to the address:

*flash memory base address* + [(*PCROP1x\_END* + 1) x 0x200] (excluded).

The minimum PCROP area size is two PCROP subpages (2 x 512 bytes):

*PCROP1x\_END* = *PCROP1x\_STRT* + 1.

When

*PCROP1x\_END* = *PCROP1x\_STRT*,

the full flash memory is PCROP-protected.

For example, to PCROP-protect the address area from 0x0800 0800 to 0x0800 13FF, set the PCROP start subpage bitfield of the FLASH\_PCROP1xSR register and the PCROP end subpage bitfield of the FLASH\_PCROP1xER register (*x* = A or B) as follows:

- PCROP1x\_STRT = 0x04 (PCROP area start address 0x0800 0800)
- PCROP1x\_END = 0x09 (PCROP area end address 0x0800 13FF)

Data read access to a PCROP-protected address raises the RDERR flag.

PCROP-protected addresses are also write protected. Write access to a PCROP-protected address raises the WRPERR flag.

PCROP-protected areas are also erase protected. Attempts to erase a page including at least one PCROP-protected subpage fails. Moreover, software mass erase cannot be performed if a PCROP-protected area is defined.

Deactivation of PCROP can only occur upon the RDP change from level 1 to level 0. Modification of user options to clear PCROP or to decrease the size of a PCROP-protected area do not have any effect to the PCROP areas. On the contrary, it is possible to increase the size of the PCROP-protected areas.

With the option bit PCROP\_RDP cleared, the change of RDP from level 1 to level 0 triggers a partial mass erase that preserves the contents of the flash memory pages overlapping with PCROP-protected areas. Refer to section [Changing the read protection level](#) for details.

**Table 22. PCROP protection**

PCROP register values (x = A or B)	PCROP-protected area
PCROP1x_STRT = PCROP1x_END	Full flash memory
PCROP1x_STRT > PCROP1x_END <sup>(1)</sup>	None (unprotected)
PCROP1x_STRT < PCROP1x_END	Subpages from PCROP1x_STRT to PCROP1x_END (read-, write-, and erase-protected); PCROP area boundary pages (erase-protected).

1. The PCROPx\_STRT and PCROPx\_END addresses cannot be pointing to the same flash page, for this comparison to work properly.

**Note:** *With PCROP\_RDP cleared, it is recommended to either define the PCROP area start and end onto flash memory page boundaries (2-Kbyte granularity), or to keep reserved and empty the PCROP-unprotected memory space of the PCROP area boundary pages (pages inside which the PCROP area starts and ends).*

### 4.5.3 FLASH write protection (WRP)

The user area in flash memory can be protected against unwanted write operations. Two write-protected (WRP) areas can be defined, with page (2-Kbyte) granularity. Each area is defined by a start page offset and an end page offset related to the physical flash memory base address. These offsets are defined in the WRP address registers [FLASH WRP area A address register \(FLASH\\_WRP1AR\)](#) and [FLASH WRP area B address register \(FLASH\\_WRP1BR\)](#).

The WRP x area (x = A, B) is defined from the address

*flash memory Base address + [WRP1x\_STRT x 0x0800] (included)*

to the address

*flash memory Base address + [(WRP1x\_END+1) x 0x0800] (excluded).*

The minimum WRP area size is one WRP page (2 Kbytes):

*WRP1x\_END = WRP1x\_STRT.*

For example, to protect the flash memory by WRP from the address 0x0800 1000 (included) to the address 0x0800 3FFF (included):

If boot in flash memory is selected, FLASH\_WRP1AR register must be programmed with:

- WRP1A\_STRT = 0x02.
- WRP1A\_END = 0x07.

WRP1B\_STRT and WRP1B\_END in FLASH\_WRP1BR can be used instead (area B in the flash memory).

When WRP is active, it cannot be erased or programmed. Consequently, a software mass erase cannot be performed if one area is write-protected.

If an erase/program operation to a write-protected part of the flash memory is attempted, the write protection error flag (WRPERR) of the FLASH\_SR register is set. This flag is also set for any write access to:

- OTP area
- part of the flash memory that can never be written like the ICP
- PCROP area

**Note:** *When the flash memory read protection level is selected (RDP level = 1), it is not possible to program or erase the memory if the CPU debug features are connected (single wire) or boot code is being executed from SRAM or system flash memory, even if WRP is not activated. Any attempt generates a hard fault (BusFault).*

**Table 23: WRP protection**

WRP registers values (x = A or B)	WRP-protected area
WRP1x_STRT = WRP1x_END	Page WRP1x
WRP1x_STRT > WRP1x_END	None (unprotected)
WRP1x_STRT < WRP1x_END	Pages from WRP1x_STRT to WRP1x_END

**Note:** *To validate the WRP options, the option bytes must be reloaded by setting the OBL\_LAUNCH bit in flash memory control register.*

#### 4.5.4 Securable memory area

The main purpose of the securable memory area is to protect a specific part of flash memory against undesired access. After system reset, the code in the securable memory area can only be executed until the securable area becomes secured and never again until the next system reset. This allows implementing software security services such as secure key storage or safe boot.

Securable memory area is located in the main flash memory. It is dedicated to executing trusted code. When not secured, the securable memory behaves like the rest of main flash memory. When secured (the SEC\_PROT bit of the FLASH\_CR register set), any access (fetch, read, programming, erase) to securable memory area is rejected, generating a bus error. The securable area can only be unsecured by a system reset.

The size of the securable memory area is defined by the SEC\_SIZE[5:0] bitfield of the FLASH\_SECR register. It can be only modified in RDP level 0 or in RDP level 1 when

SEC\_PROT = 0. Its content is erased upon changing from RDP level 1 to level 0, even if it overlaps with PCROP subpages.

*Note:* The securable memory area start address is 0x0800 0000. Before activating the securable memory area, move the vector table outside the page 0 if necessary.

*Note:* Upon change from RDP level 1 to level 0 while the PCROP\_RDP bit is cleared, the securable memory area is erased even if it overlaps with the PCROP subpages. The PCROP subpages not overlapping with the securable memory area are not erased. See [Table 24](#).

**Table 24. Securable memory erase at RDP level 1 to level 0 change**

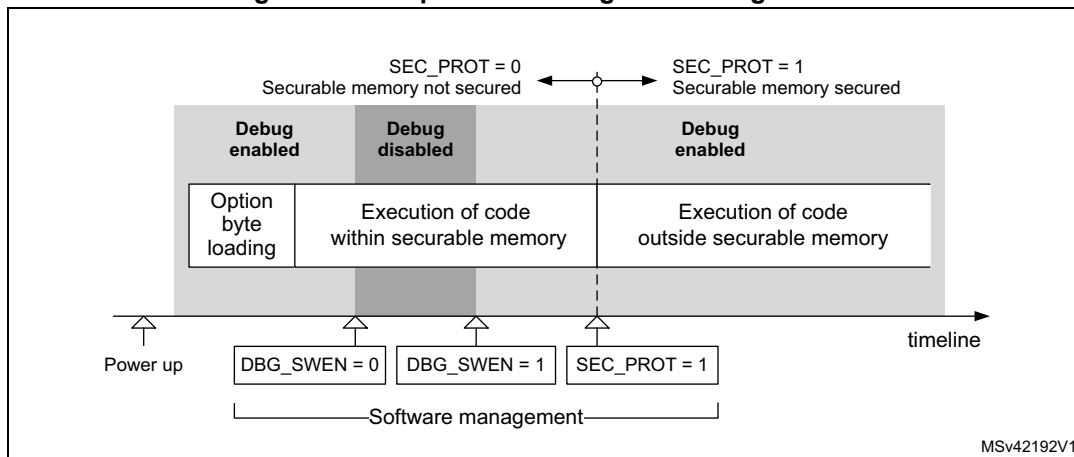
Securable memory size (SEC_SIZE[4:0])	PCROP_RDP	Erased pages
0	1	All (mass erase)
0	0	All but PCROP
> 0	1	All (mass erase)
> 0	0	All but PCROP outside the securable memory area

#### 4.5.5 Disabling core debug access

For executing sensitive code or manipulating sensitive data in securable memory area, the debug access to the core can temporarily be disabled.

[Figure 4](#) gives an example of managing DBG\_SWEN and SEC\_PROT bits.

**Figure 4. Example of disabling core debug access**



#### 4.5.6 Forcing boot from main flash memory

To increase the security and establish a chain of trust, the BOOT\_LOCK option bit of the FLASH\_SECR register allows forcing the system to boot from the main flash memory

regardless of the other boot options. It is always possible to set the BOOT\_LOCK bit. However, it is possible to reset it 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.

**Caution:** If BOOT\_LOCK is set in association with RDP level 1, the debug capabilities of the device are disabled and the reset value of the DBG\_SWEN bit of the FLASH\_ACR register becomes zero. If DBG\_SWEN bit is not set by the application code after reset, there is no way to recover from this situation.

## 4.6 FLASH interrupts

Table 25. FLASH interrupt requests

Interrupt event	Event flag	Event flag/interrupt clearing method	Interrupt enable control bit
End of operation	EOP <sup>(1)</sup>	Write EOP=1	EOPIE
Operation error	OPERR <sup>(2)</sup>	Write OPERR=1	ERRIE
Read protection error	RDERR	Write RDERR=1	RDERRIE
Write protection error	WRPERR	Write WRPERR=1	N/A
Size error	SIZERR	Write SIZERR=1	N/A
Programming sequential error	PROGERR	Write PROGERR=1	N/A
Programming alignment error	PGAERR	Write PGAERR=1	N/A
Programming sequence error	PGSERR	Write PGSERR=1	N/A
Data miss during fast programming error	MISSERR	Write MISSERR=1	N/A
Fast programming error	FASTERR	Write FASTERR=1	N/A

1. EOP is set only if EOPIE is set.

2. OPERR is set only if ERRIE is set.

## 4.7 FLASH registers

### 4.7.1 FLASH access control register (FLASH\_ACR)

Address offset: 0x000

Reset value: 0b0000 0000 0000 010X 0000 0110 0000 0000 (the EMPTY bit is updated only by OBL. It is not affected by the system reset.)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_SWEN	Res.	EMPTY
													rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	ICRST	Res.	ICEN	PRFTEN	Res.	Res.	Res.	Res.	Res.	LATENCY[2:0]		
				rw		rw	rw						rw	rw	rw

Bits 31:19 Reserved, must be kept at reset value.

Bit 18 **DBG\_SWEN**: Debug access software enable

Software may use this bit to enable/disable the debugger read access.

0: Debugger disabled

1: Debugger enabled

Bit 17 Reserved, must be kept at reset value.

Bit 16 **EMPTY**: Main flash memory area empty

This bit indicates whether the first location of the main flash memory area was read as erased or as programmed during OBL. It is not affected by the system reset. Software may need to change this bit value after a flash memory program or erase operation.

0: Main flash memory area programmed

1: Main flash memory area empty

The bit can be set and reset by software.

Bits 15:12 Reserved, must be kept at reset value.

Bit 11 **ICRST**: CPU Instruction cache reset

0: CPU Instruction cache is not reset

1: CPU Instruction cache is reset

This bit can be written only when the instruction cache is disabled.

Bit 10 Reserved, must be kept at reset value.

Bit 9 **ICEN**: CPU Instruction cache enable

0: CPU Instruction cache is disabled

1: CPU Instruction cache is enabled

Bit 8 **PRFTEN**: CPU Prefetch enable

0: CPU Prefetch disabled

1: CPU Prefetch enabled

Bits 7:3 Reserved, must be kept at reset value.

Bits 2:0 **LATENCY[2:0]**: Flash memory access latency

The value in this bitfield represents the number of CPU wait states when accessing the flash memory.

000: Zero wait states

001: One wait state

Other: Reserved

A new write into the bitfield becomes effective when it returns the same value upon read.

## 4.7.2 FLASH key register (FLASH\_KEYR)

Address offset: 0x008

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:0 **KEY[31:0]**: FLASH key

The following values must be written consecutively to unlock the *FLASH control register (FLASH\_CR)*, thus enabling programming/erasing operations:

KEY1: 0x4567 0123

KEY2: 0xCDEF 89AB

## 4.7.3 FLASH option key register (FLASH\_OPTKEYR)

Address offset: 0x00C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OPTKEY[31:16]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTKEY[15:0]															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bits 31:0 **OPTKEY[31:0]**: Option byte key

The following values must be written consecutively to unlock the flash memory option registers, enabling option byte programming/erasing operations:

KEY1: 0x0819 2A3B

KEY2: 0x4C5D 6E7F

#### 4.7.4 FLASH status register (FLASH\_SR)

Address offset: 0x010

Reset value: 0x000X 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFGBSY	Res.	BSY1
													r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTV ERR	RD ERR	Res.	Res.	Res.	Res.	FAST ERR	MISS ERR	PGS ERR	SIZ ERR	PGA ERR	WRP ERR	PROG ERR	Res.	OP ERR	EOP
rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

Bits 31:19 Reserved, must be kept at reset value.

Bit 18 **CFGBSY**: Programming or erase configuration busy.

This flag is set and reset by hardware.

For flash program operation, it is set when the first word is sent, and cleared after the second word is sent when the operation completes or ends with an error.

For flash erase operation, it is set when setting the STRT bit of the FLASH\_CR register and cleared when the operation completes or ends with an error.

When set, a programming or erase operation is ongoing and the corresponding settings in the [FLASH control register \(FLASH\\_CR\)](#) are used (busy) and cannot be changed. Any other flash operation launch must be postponed.

When cleared, the programming and erase settings in the [FLASH control register \(FLASH\\_CR\)](#) can be modified.

*Note: The CFGBSY bit is also set when attempting to write locked flash memory (with the first byte sent). When the CFGBSY bit is set, writing into the FLASH\_CR register causes HardFault. To clear the CFGBSY bit, send a double word to the flash memory and wait until the access is finished (otherwise the CFGBSY bit remains set).*

Bit 17 Reserved, must be kept at reset value.

Bit 16 **BSY1**: Busy

This flag indicates that a flash memory operation requested by [FLASH control register \(FLASH\\_CR\)](#) is in progress. This bit is set at the beginning of the flash memory operation, and cleared when the operation finishes or when an error occurs.

Bit 15 **OPTVERR**: Option and Engineering bits loading validity error

Set by hardware when the options and engineering bits read may not be the one configured by the user or production. If options and engineering bits haven't been properly loaded, OPTVERR is set again after each system reset. Option bytes that fail loading are forced to a safe value, see [Section 4.4.2: FLASH option byte programming](#).

Cleared by writing 1.

Bit 14 **RDERR**: PCROP read error

Set by hardware when an address to be read belongs to a read protected area of the flash memory (PCROP protection). An interrupt is generated if RDERRIE is set in FLASH\_CR.

Cleared by writing 1.

Bits 13:10 Reserved, must be kept at reset value.



- Bit 9 **FASTERR**: Fast programming error  
Set by hardware when a fast programming sequence (activated by FSTPG) is interrupted due to an error (alignment, size, write protection or data miss). The corresponding status bit (PGAERR, SIZERR, WRPERR or MISSERR) is set at the same time.  
Cleared by writing 1.
- Bit 8 **MISSERR**: Fast programming data miss error  
In Fast programming mode, 32 double words (256 bytes) must be sent to flash memory successively, and the new data must be sent to the logic control before the current data is fully programmed. MISSERR is set by hardware when the new data is not present in time.  
Cleared by writing 1.
- Bit 7 **PGSERR**: 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.  
Cleared by writing 1.
- Bit 6 **SIZERR**: 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 (consequently: word access).  
Cleared by writing 1.
- Bit 5 **PGAERR**: 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.  
Cleared by writing 1.
- Bit 4 **WRPERR**: 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.  
Cleared by writing 1.
- Bit 3 **PROGERR**: 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'.  
Cleared by writing 1.
- Bit 2 Reserved, must be kept at reset value.
- Bit 1 **OPERR**: Operation error  
Set by hardware when a flash memory operation (program / erase) completes unsuccessfully.  
This bit is set only if error interrupts are enabled (ERRIE=1).  
Cleared by writing '1'.
- Bit 0 **EOP**: End of operation  
Set by hardware when one or more flash memory operation (programming / erase) has been completed successfully.  
This bit is set only if the end of operation interrupts are enabled (EOPIE=1).  
Cleared by writing 1.

### 4.7.5 FLASH control register (FLASH\_CR)

Address offset: 0x014

Reset value: 0xC000 0000

Access: no wait state when no flash memory operation is on going, word, half-word and byte access

This register must not be modified when CFGBSY in *FLASH status register (FLASH\_SR)* is set. This would result in a HardFault exception.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPT LOCK	Res.	SEC PROT	OBL LAUNCH	RD ERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPT STRT	STRT
rs	rs		rw	rc_w1	rw	rw	rw						rw	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	PNB[6:0]							MER1	PER	PG
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

**Bit 31 LOCK:** FLASH\_CR Lock

This bit is set only. When set, the FLASH\_CR register is locked. It is cleared by hardware after detecting the unlock sequence.

In case of an unsuccessful unlock operation, this bit remains set until the next system reset.

**Bit 30 OPTLOCK:** Options Lock

This bit is set only. When set, all bits concerning user option in FLASH\_CR register and so option page are locked. This bit is cleared by hardware after detecting the unlock sequence. The LOCK bit must be cleared before doing the unlock sequence for OPTLOCK bit.

In case of an unsuccessful unlock operation, this bit remains set until the next reset.

Bit 29 Reserved, must be kept at reset value.

**Bit 28 SEC\_PROT:** Securable memory area protection enable

This bit enables the protection on securable area, provided that a non-null securable memory area size (SEC\_SIZE[4:0]) is defined in option bytes.

0: Disable (securable area accessible)

1: Enable (securable area not accessible)

This bit is possible to set only by software and to clear only through a system reset.

**Bit 27 OBL\_LAUNCH:** Option byte load launch

When set, this bit triggers the load of option bytes into option registers. It is automatically cleared upon the completion of the load. The high state of the bit indicates pending option byte load.

The bit cannot be cleared by software. It cannot be written as long as OPTLOCK is set.

**Bit 26 RDERRIE:** PCROP read error interrupt enable

This bit enables the interrupt generation upon setting the RDERR flag in the FLASH\_SR register.

0: Disable

1: Enable

**Bit 25 ERRIE:** Error interrupt enable

This bit enables the interrupt generation upon setting the OPERR flag in the FLASH\_SR register.

0: Disable

1: Enable

- Bit 24 **EOPIE**: End-of-operation interrupt enable  
This bit enables the interrupt generation upon setting the EOP flag in the FLASH\_SR register.  
0: Disable  
1: Enable
- Bits 23:19 Reserved, must be kept at reset value.
- Bit 18 **FSTPG**: Fast programming enable  
0: Disable  
1: Enable
- Bit 17 **OPTSTRT**: Start of modification of option bytes  
This bit triggers an options operation when set.  
This bit is set only by software, and is cleared when the BSY1 bit is cleared in FLASH\_SR.
- Bit 16 **STRT**: Start erase operation  
This bit triggers an erase operation when set.  
This bit is possible to set only by software and to clear only by hardware. The hardware clears it when one of BSY1 and BSY2 flags in the FLASH\_SR register transits to zero.
- Bits 15:10 Reserved, must be kept at reset value.
- Bits 9:3 **PNB[6:0]**: Page number selection  
These bits select the page to erase:  
0x00: page 0  
0x01: page 1  
...  
0x7F: page 127  
*Note: Values corresponding to addresses outside the main flash memory are not allowed.  
See [Table 9](#) and [Table 10](#).*
- Bit 2 **MER1**: Mass erase  
When set, this bit triggers the mass erase, that is, all user pages.
- Bit 1 **PER**: Page erase enable  
0: Disable  
1: Enable
- Bit 0 **PG**: Flash memory programming enable  
0: Disable  
1: Enable

#### 4.7.6 FLASH option register (FLASH\_OPTR)

Address offset: 0x020

Reset value: 0xFFFF XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word and byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FDCAN_BL_CK[1:0]		IRHEN	NRST_MODE [1:0]		N BOOT0	N BOOT1	NBOOT_SEL	SECURE_MUXING_EN	RAM_PARITY_CHECK	HSE_NOR_REMAPPED	Res.	WWDG_SW	IWDG_STDBY	IWDG_STOP	IWDG_SW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NRST_SHDW	NRST_STDBY	NRST_STOP	BORF_LEV[1:0]		BORR_LEV[1:0]		BOR_EN	RDP[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:30 **FDCAN\_BL\_CK[1:0]**: FDCAN bootloader clock source

- 00: HSI48
- 01: HSE crystal - 12 MHz
- 10: HSE crystal - 24 MHz
- 11: HSE crystal - 48 MHz

*Note: Only available on STM32C092xx devices, reserved on the other products.*

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]**: PF2-NRST pin configuration

- 00: Reserved
- 01: Reset input only: a low level on the NRST pin generates system reset; internal RESET is not propagated to the NRST pin.
- 10: PF2 GPIO: only internal RESET is possible
- 11: Bidirectional reset: the NRST pin is configured in reset input/output (legacy) mode

Bit 26 **NBOOT0**: NBOOT0 option bit

- 0: NBOOT0 = 0
- 1: NBOOT0 = 1

Bit 25 **NBOOT1**: NBOOT1 boot configuration

Together with the BOOT0 pin or NBOOT0 option bit (depending on NBOOT\_SEL option bit configuration), this bit selects boot mode from the main flash memory, SRAM, or the system memory. Refer to [Section 3: Boot modes](#).

Bit 24 **NBOOT\_SEL**: BOOT0 signal source selection

- This bit defines the source of the BOOT0 signal.
- 0: BOOT0 pin (legacy mode)
- 1: NBOOT0 option bit

- Bit 23 **SECURE\_MUXING\_EN**: Multiple-bonding security  
The bit allows enabling automatic I/O configuration to prevent conflicts on I/Os connected (bonded) onto the same pin.  
0: Disable  
1: Enable  
If the software sets one of the I/Os connected to the same pin as active by configuring the SYSCFG\_CFGR3 register, enabling this bit automatically forces the other I/Os in digital input mode, regardless of their software configuration.  
When the bit is disabled, the SYSCFG\_CFGR3 register setting is ignored, all GPIOs linked to a given pin are active and can be set in the mode specified by the corresponding GPIOx\_MODER register. The user software must ensure that there is no conflict between GPIOs.
- Bit 22 **RAM\_PARITY\_CHECK**: SRAM parity check control enable/disable  
0: Enable  
1: Disable
- Bit 21 **HSE\_NOT\_REMAPPED**: HSE remapping enable/disable  
When cleared, the bit remaps the HSE clock source from PF0-OSC\_IN/PF1-OSC\_OUT pins to PC14-OSCX\_IN/PC15-OSCX\_OUT. Thus PC14-OSCX\_IN/PC15-OSCX\_OUT are shared by both LSE and HSE and the two clock sources cannot be use simultaneously.  
0: Enable  
1: Disable  
On packages with less than 48 pins, the remapping is always enabled (PF0-OSC\_IN/PF1-OSC\_OUT are not available), regardless of this bit. As all STM32C011xx packages have less than 48 pins, this bit is only applicable to STM32C031xx.  
*Note: On 48 pins packages, when HSE\_NOT\_REMAPPED is reset, HSE cannot be used in bypass mode. Refer to product errata sheet for more details.*
- Bit 20 Reserved, must be kept at reset value.
- Bit 19 **WWDG\_SW**: Window watchdog selection  
0: Hardware window watchdog  
1: Software window watchdog
- Bit 18 **IWDG\_STDBY**: Independent watchdog counter freeze in Standby mode  
0: Independent watchdog counter is frozen in Standby mode  
1: Independent watchdog counter is running in Standby mode
- Bit 17 **IWDG\_STOP**: Independent watchdog counter freeze in Stop mode  
0: Independent watchdog counter is frozen in Stop mode  
1: Independent watchdog counter is running in Stop mode
- Bit 16 **IWDG\_SW**: Independent watchdog selection  
0: Hardware independent watchdog  
1: Software independent watchdog
- Bit 15 **NRST\_SHDW**: Reset generation upon entering Shutdown mode  
0: Reset generated  
1: Reset not generated
- Bit 14 **NRST\_STDBY**: Reset generation upon entering Standby mode  
0: Reset generated  
1: Reset not generated
- Bit 13 **NRST\_STOP**: Reset generation upon entering Stop mode  
0: Reset generated  
1: Reset not generated

Bits 12:11 **BORF\_LEV[1:0]**: BOR threshold at falling  $V_{DD}$  supply

Falling  $V_{DD}$  crossings this threshold activates the reset signal.

00: BOR falling level 1 with threshold around 2.0 V

01: BOR falling level 2 with threshold around 2.2 V

10: BOR falling level 3 with threshold around 2.5 V

11: BOR falling level 4 with threshold around 2.8 V

Bits 10:9 **BORR\_LEV[1:0]**: BOR threshold at rising  $V_{DD}$  supply

Rising  $V_{DD}$  crossings this threshold releases the reset signal.

00: BOR rising level 1 with threshold around 2.1 V

01: BOR rising level 2 with threshold around 2.3 V

10: BOR rising level 3 with threshold around 2.6 V

11: BOR rising level 4 with threshold around 2.9 V

Bit 8 **BOR\_EN**: Brown out reset enable

0: Configurable brown out reset disabled, power-on reset defined by POR/PDR levels

1: Configurable brown out reset enabled, values of BORR\_LEV and BORF\_LEV taken into account

Bits 7:0 **RDP[7:0]**: Read protection level

0xAA: Level 0, read protection not active

0xCC: Level 2, chip read protection active

Other: Level 1, memories read protection active

#### 4.7.7 FLASH PCROP area A start address register (FLASH\_PCROP1ASR)

Address offset: 0x024

Reset value: 0b0000 0000 0000 0000 0000 000X XXXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_STRT[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **PCROP1A\_STRT[8:0]**: PCROP1A area start offset

Contains the offset of the first subpage of the PCROP1A area.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

#### 4.7.8 FLASH PCROP area A end address register (FLASH\_PCROP1AER)

Address offset: 0x028

Reset value: 0bX000 0000 0000 0000 0000 000X XXXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word access. PCROP\_RDP bit can be accessed with byte access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1A_END[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 **PCROP\_RDP**: PCROP area erase upon RDP level regression

This bit determines whether the PCROP area (and the totality of the PCROP area boundary pages) is erased by the mass erase triggered by the RDP level regression from level 1 to level 0:

0: Not erased

1: Erased

The software can only set this bit. It is automatically reset upon mass erase following the RDP regression from level 1 to level 0.

Bits 30:9 Reserved, must be kept at reset value.

Bits 8:0 **PCROP1A\_END[8:0]**: PCROP1A area end offset

Contains the offset of the last subpage of the PCROP1A area.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

#### 4.7.9 FLASH WRP area A address register (FLASH\_WRP1AR)

Address offset: 0x02C

Reset value: 0b0000 0000 0XXX XXXX 0000 0000 0XXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[6:0]						
									rw	rw	rw	rw	rw	rw	rw

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP1A\_END[6:0]**: WRP area A end offset

This bitfield contains the offset of the last page of the WRP area A.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP1A\_STRT[6:0]**: WRP area A start offset

This bitfield contains the offset of the first page of the WRP area A.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

#### 4.7.10 FLASH WRP area B address register (FLASH\_WRP1BR)

Address offset: 0x030

Reset value: 0b0000 0000 0XXX XXXX 0000 0000 0XXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word and byte access.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[6:0]						
									rw	rw	rw	rw	rw	rw	rw

Bits 31:23 Reserved, must be kept at reset value.

Bits 22:16 **WRP1B\_END[6:0]**: WRP area B end offset

This bitfield contains the offset of the last page of the WRP area B.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **WRP1B\_STRT[6:0]**: WRP area B start offset

This bitfield contains the offset of the first page of the WRP area B.

*Note: The number of effective bits depends on the size of the flash memory in the device.*



#### 4.7.11 FLASH PCROP area B start address register (FLASH\_PCROP1BSR)

Address offset: 0x034

Reset value: 0b0000 0000 0000 0000 0000 000X XXXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_STRT[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **PCROP1B\_STRT[8:0]**: PCROP1B area start offset

Contains the offset of the first subpage of the PCROP1B area.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

#### 4.7.12 FLASH PCROP area B end address register (FLASH\_PCROP1BER)

Address offset: 0x038

Reset value: 0b0000 0000 0000 0000 0000 000X XXXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1B_END[8:0]								
							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bits 31:9 Reserved, must be kept at reset value.

Bits 8:0 **PCROP1B\_END[8:0]**: PCROP1B area end offset

Contains the offset of the last subpage of the PCROP1B area.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

### 4.7.13 FLASH security register (FLASH\_SECR)

Address offset: 0x080

Reset value: 0b0000 0000 0000 000X 0000 0000 0XXX XXXX (The option bits are loaded with values from flash memory at power-on reset release.)

Access: no wait state when no flash memory operation is on going, word, half-word access

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT_LOCK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC_SIZE[6:0]						
									rw	rw	rw	rw	rw	rw	rw

Bits 31:17 Reserved, must be kept at reset value.

Bit 16 **BOOT\_LOCK**: used to force boot from user area

0: Boot based on the pad/option bit configuration

1: Boot forced from main flash memory

**Caution:** If the bit is set in association with RDP level 1, the debug capabilities of the device are disabled and the reset value of the DBG\_SWEN bit of the FLASH\_ACR register becomes zero. In this case, re-enabling of debug capabilities is possible only by setting the DBG\_SWEN bit by the application code.

Bits 15:7 Reserved, must be kept at reset value.

Bits 6:0 **SEC\_SIZE[6:0]**: Securable memory area size

Contains the number of securable flash memory pages.

*Note: The number of effective bits depends on the size of the flash memory in the device.*

## 4.7.14 FLASH register map

Table 26. FLASH register map and reset values

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x000	FLASH_ACR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DBG_SWEN	Res	EMPTY	Res	Res	Res	Res	ICRST	Res	ICEN	PRFTEN	Res	Res	Res	Res	Res	LATENCY [2:0]			
	Reset value														1		X					0		1	0						0	0	0	
0x004	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x008	FLASH_KEYR	KEYR[31:0]																																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x00C	FLASH_OPT_KEYR	OPTKEY[31:0]																																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x010	FLASH_SR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CFGBSY	Res	BSY1	OPTVERR	RDERR	Res	Res	Res	Res	Res	FASTERR	MISERR	PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res	OPERR	EOP
	Reset value														0		0	X	0					0	0	0	0	0	0	0		0	0	
0x014	FLASH_CR	LOCK	OPTLOCK	Res	SEC_PROT	OBL_LAUNCH	RDERRIE	ERRIE	EOPIE	Res	Res	Res	Res	Res	FSTPG	OPTSTRT	STRT	Res	Res	Res	Res	Res	Res	PNB[6:0]						MER1	PER	PG		
	Reset value	1	1		0	0	0	0	0						0	0	0							0	0	0	0	0	0	0	0	0	0	0
0x018	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
0x020	FLASH_OPTR	FDCAN_BLK[1:0]	IRHEN	NRST_MODE[1:0]		NBOOT0	NBOOT1	NBOOT_SEL	SECURE_MUXING_EN	RAM_PARITY_CHECK	HSE_NOT_REMAPPED	Res	WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	NRST_SHDW	NRST_STDBY	NRST_STOP	BORF_LEV[1:0]	BORR_LEV[1:0]	BOR_EN	RDP[7:0]											
	Reset value	X	X	X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
0x024	FLASH_PCROP1ASR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PCROP1A_STRT[8:0]									
	Reset value																								X	X	X	X	X	X	X	X	X	
0x028	FLASH_PCROP1AER	PCROP_RDP	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PCROP1A_END[8:0]									
	Reset value	X																							X	X	X	X	X	X	X	X	X	
0x02C	FLASH_WRP1AR	Res	Res	Res	Res	Res	Res	Res	Res	Res	WRP1A_END[6:0]						Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	Reset value										X	X	X	X	X	X	X																	
0x030	FLASH_WRP1BR	Res	Res	Res	Res	Res	Res	Res	Res	Res	WRP1B_END[6:0]						Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	
	Reset value										X	X	X	X	X	X	X																	
0x034	FLASH_PCROP1BSR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PCROP1B_STRT[8:0]									
	Reset value																								X	X	X	X	X	X	X	X	X	
0x038	FLASH_PCROP1BER	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PCROP1B_END[8:0]									
	Reset value																								X	X	X	X	X	X	X	X	X	
0x03C - 0x07F	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	

Table 26. FLASH register map and reset values (continued)

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
0x080	FLASH_SECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT_LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEC_SIZE[6:0]									
	Reset value																X										X	X	X	X	X	X	X			

Refer to [Section 2.2 on page 45](#) for the register boundary addresses.