

2 Memory and bus architecture

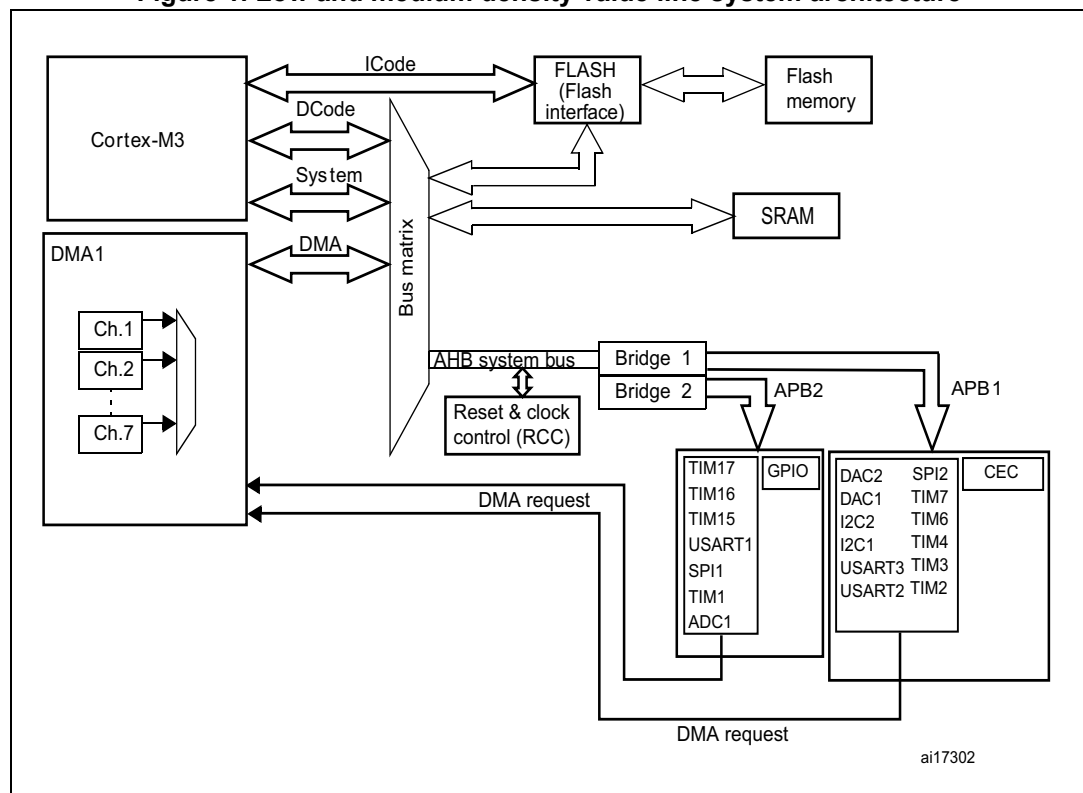
2.1 System architecture

In low-and medium-density value line devices, the main system consists of:

- Three masters:
 - Cortex[®]-M3 core DCode bus (D-bus) and System bus (S-bus)
 - GP-DMA1 (general-purpose DMA)
- Three slaves:
 - Internal SRAM
 - Internal flash memory
 - AHB to APB bridges (AHB to APBx), which connect all the APB peripherals

These are interconnected using a multilayer AHB bus architecture as shown in [Figure 1](#).

Figure 1. Low and medium density value line system architecture

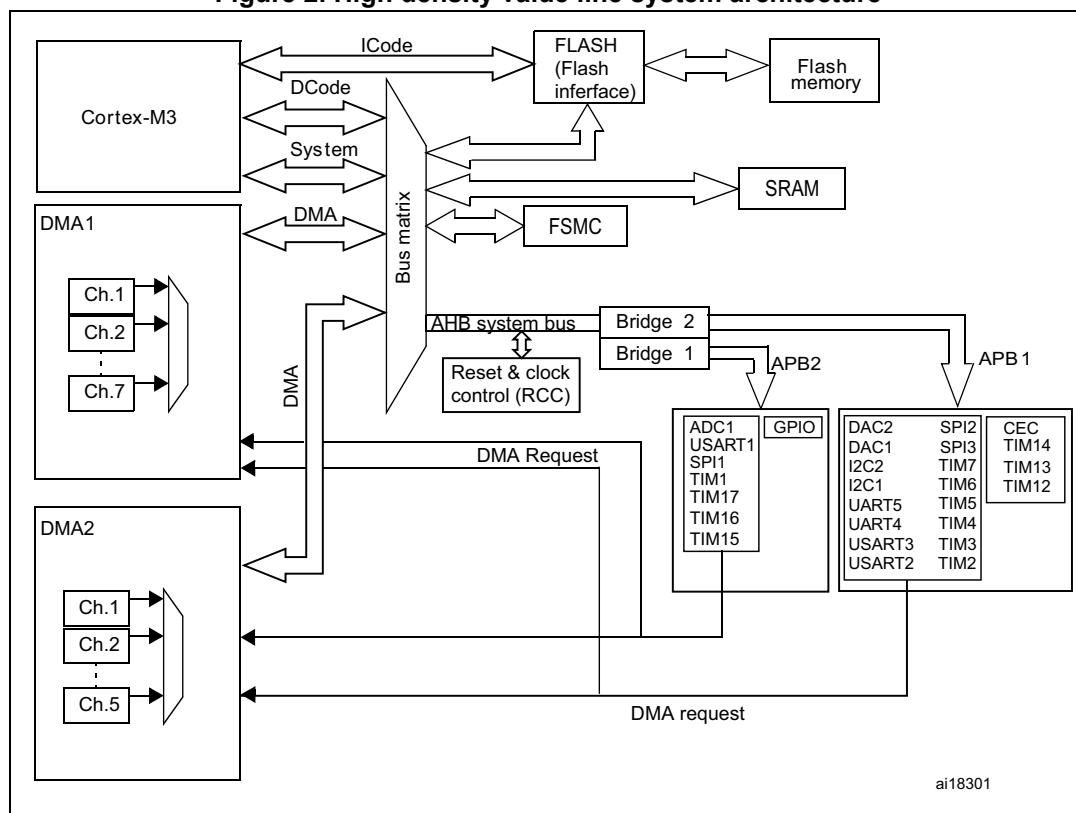


In high-density value line devices, the main system consists of:

- Four masters:
 - Cortex®-M3 core DCode bus (D-bus) and System bus (S-bus)
 - GP-DMA1 & 2 (general-purpose DMA)
- Four slaves:
 - Internal SRAM
 - Internal flash memory
 - FSMC
 - AHB to APB bridges (AHB to APBx), which connect all the APB peripherals

These are interconnected using a multilayer AHB bus architecture as shown in [Figure 2](#).

Figure 2. High density value line system architecture



ICode bus

This bus connects the instruction bus of the Cortex®-M3 core to the flash memory instruction interface. Instruction fetches are performed on this bus.

DCode bus

This bus connects the DCode bus (literal load and debug access) of the Cortex®-M3 core to the flash memory data interface.

System bus

This bus connects the system bus of the Cortex[®]-M3 core (peripherals bus) to a bus matrix which manages the arbitration between the core and the DMA.

DMA bus

This bus connects the AHB master interface of the DMA to the bus matrix which manages the access of CPU DCode and DMA to the SRAM, flash memory and peripherals.

Bus matrix

The bus matrix manages the access arbitration between the core system bus and the DMA master bus. The arbitration uses a round robin algorithm. In low and medium-density value line devices the bus matrix is composed of three masters (CPU DCode, System bus, DMA1 bus) and three slaves (FLITF, SRAM and AHB to APB bridges).

In high-density value line devices the bus matrix is composed of four masters (CPU DCode, System bus, DMA1 bus and DMA2 bus) and four slaves (FLITF, SRAM, FSMC and AHB to APB bridges).

AHB peripherals are connected to the system bus through the bus matrix to allow DMA access.

AHB/APB bridges (APB)

The two AHB/APB bridges provide full synchronous connections between the AHB and the two APB buses. APB buses operate at full speed (up to 24 MHz).

Refer to [Table 2](#) for the address mapping of the peripherals connected to each bridge.

After each device reset, all peripheral clocks are disabled (except for the SRAM and FLITF). Before using a peripheral its clock in the RCC_AHBENR, RCC_APB2ENR or RCC_APB1ENR register must be enabled.

Note: When a 16- or 8-bit access is performed on an APB register, the access is transformed into a 32-bit access: the bridge duplicates the 16- or 8-bit data to feed the 32-bit vector.

2.2 Memory organization

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbyte address space.

The bytes are coded in memory in little endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte, the most significant.

For the detailed mapping of peripheral registers, refer to the related sections.

The addressable memory space is divided into 8 main blocks, each of 512 MB.

All the memory areas that are not allocated to on-chip memories and peripherals are considered "reserved"). Refer to the memory map figure in the corresponding product datasheet.

2.3 Memory map

See the datasheet corresponding to the used device for a comprehensive diagram of the memory map. [Table 1](#) and [Table 2](#) give the boundary addresses of the peripherals available in all STM32F100xx devices.

Table 1. Low and medium-density device register boundary addresses

Boundary address	Peripheral	Bus	Register map
0x4002 3000 - 0x4002 33FF	CRC	AHB	Section 3.4.4 on page 49
0x4002 2400 - 0x4002 2FFF	Reserved		-
0x4002 2000 - 0x4002 23FF	Flash memory interface		-
0x4002 1400 - 0x4002 1FFF	Reserved		-
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC		Section 6.3.12 on page 101
0x4002 0400 - 0x4002 0FFF	Reserved		-
0x4002 0000 - 0x4002 03FF	DMA1		Section 9.4.7 on page 159
0x4001 4C00 - 0x4001 FFFF	Reserved	APB2	-
0x4001 4800 - 0x4001 4BFF	TIM17 timer		Section 15.6.16 on page 454
0x4001 4400 - 0x4001 47FF	TIM16 timer		Section 15.6.16 on page 454
0x4001 4000 - 0x4001 43FF	TIM15 timer		Section 15.5.18 on page 434
0x4001 3C00 - 0x4001 3FFF	Reserved		-
0x4001 3800 - 0x4001 3BFF	USART1		Section 23.6.8 on page 646
0x4001 3400 - 0x4001 37FF	Reserved		-
0x4001 3000 - 0x4001 33FF	SPI1		Section 21.4.8 on page 565
0x4001 2C00 - 0x4001 2FFF	TIM1 timer		Section 12.4.21 on page 282
0x4001 2800 - 0x4001 2BFF	Reserved		-
0x4001 2400 - 0x4001 27FF	ADC1		Section 10.11.15 on page 188
0x4001 1C00 - 0x4001 23FF	Reserved		-
0x4001 1800 - 0x4001 1BFF	GPIO Port E		Section 7.5 on page 130
0x4001 1400 - 0x4001 17FF	GPIO Port D		Section 7.5 on page 130
0x4001 1000 - 0x4001 13FF	GPIO Port C		Section 7.5 on page 130
0x4001 0C00 - 0x4001 0FFF	GPIO Port B		Section 7.5 on page 130
0x4001 0800 - 0x4001 0BFF	GPIO Port A		Section 7.5 on page 130
0x4001 0400 - 0x4001 07FF	EXTI		Section 8.3.7 on page 143
0x4001 0000 - 0x4001 03FF	AFIO		Section 7.5 on page 130

Table 1. Low and medium-density device register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 FFFF	Reserved	APB1	-
0x4000 7800 - 0x4000 7BFF	CEC		Section 24.9.8 on page 668
0x4000 7400 - 0x4000 77FF	DAC		Section 11.5.15 on page 210
0x4000 7000 - 0x4000 73FF	Power control PWR		Section 4.4.3 on page 63
0x4000 6C00 - 0x4000 6FFF	Backup registers (BKP)		Section 5.4.5 on page 69
0x4000 5C00 - 0x4000 6BFF	Reserved		-
0x4000 5800 - 0x4000 5BFF	I2C2		Section 22.6.10 on page 598
0x4000 5400 - 0x4000 57FF	I2C1		Section 22.6.10 on page 598
0x4000 4C00 - 0x4000 53FF	Reserved		-
0x4000 4800 - 0x4000 4BFF	USART3		Section 23.6.8 on page 646
0x4000 4400 - 0x4000 47FF	USART2		Section 23.6.8 on page 646
0x4000 3C00 - 0x4000 3FFF	Reserved		-
0x4000 3800 - 0x4000 3BFF	SPI2		Section 21.4.8 on page 565
0x4000 3400 - 0x4000 37FF	Reserved		-
0x4000 3000 - 0x4000 33FF	Independent watchdog (IWDG)		Section 18.4.5 on page 486
0x4000 2C00 - 0x4000 2FFF	Window watchdog (WWDG)		Section 19.6.4 on page 493
0x4000 2800 - 0x4000 2BFF	RTC		Section 17.4.7 on page 480
0x4000 1800 - 0x4000 27FF	Reserved		-
0x4000 1400 - 0x4000 17FF	TIM7 timer		Section 16.4.9 on page 468
0x4000 1000 - 0x4000 13FF	TIM6 timer		Section 16.4.9 on page 468
0x4000 0C00 - 0x4000 0FFF	Reserved		-
0x4000 0800 - 0x4000 0BFF	TIM4 timer		Section 13.4.19 on page 340
0x4000 0400 - 0x4000 07FF	TIM3 timer		Section 13.4.19 on page 340
0x4000 0000 - 0x4000 03FF	TIM2 timer		Section 13.4.19 on page 340

Table 2. High-density device register boundary addresses

Boundary address	Peripheral	Bus	Register map
0x4002 3000 - 0x4002 33FF	CRC	AHB	Section 3.4.4 on page 49
0x4002 2400 - 0x4002 2FFF	Reserved		-
0x4002 2000 - 0x4002 23FF	Flash memory interface		-
0x4002 1400 - 0x4002 1FFF	Reserved		-
0x4002 1000 - 0x4002 13FF	Reset and clock control RCC		Section 6.3.12 on page 101
0x4002 0800 - 0x4002 0FFF	Reserved		-
0x4002 0400 - 0x4002 07FF	DMA2		Section 9.4.7 on page 159
0x4002 0000 - 0x4002 03FF	DMA1		Section 9.4.7 on page 159

Table 2. High-density device register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4001 4C00 - 0x4001 FFFF	Reserved	APB2	-
0x4001 4800 - 0x4001 4BFF	TIM17 timer		Section 15.6.16 on page 454
0x4001 4400 - 0x4001 47FF	TIM16 timer		Section 15.6.16 on page 454
0x4001 4000 - 0x4001 43FF	TIM15 timer		Section 15.5.18 on page 434
0x4001 3C00 - 0x4001 3FFF	Reserved		-
0x4001 3800 - 0x4001 3BFF	USART1		Section 23.6.8 on page 646
0x4001 3400 - 0x4001 37FF	Reserved		-
0x4001 3000 - 0x4001 33FF	SPI1		Section 21.4.8 on page 565
0x4001 2C00 - 0x4001 2FFF	TIM1 timer		Section 12.4.21 on page 282
0x4001 2800 - 0x4001 2BFF	Reserved		-
0x4001 2400 - 0x4001 27FF	ADC1		Section 10.11.15 on page 188
0x4001 2000 - 0x4001 23FF	GPIO Port G		Section 7.5 on page 130
0x4001 1C00 - 0x4001 1FFF	GPIO Port F		Section 7.5 on page 130
0x4001 1800 - 0x4001 1BFF	GPIO Port E		Section 7.5 on page 130
0x4001 1400 - 0x4001 17FF	GPIO Port D		Section 7.5 on page 130
0x4001 1000 - 0x4001 13FF	GPIO Port C		Section 7.5 on page 130
0x4001 0C00 - 0x4001 0FFF	GPIO Port B		Section 7.5 on page 130
0x4001 0800 - 0x4001 0BFF	GPIO Port A		Section 7.5 on page 130
0x4001 0400 - 0x4001 07FF	EXTI		Section 8.3.7 on page 143
0x4001 0000 - 0x4001 03FF	AFIO		Section 7.5 on page 130

Table 2. High-density device register boundary addresses (continued)

Boundary address	Peripheral	Bus	Register map
0x4000 7C00 - 0x4000 FFFF	Reserved	APB1	-
0x4000 7800 - 0x4000 7BFF	CEC		Section 24.9.8 on page 668
0x4000 7400 - 0x4000 77FF	DAC		Section 11.5.15 on page 210
0x4000 7000 - 0x4000 73FF	Power control PWR		Section 4.4.3 on page 63
0x4000 6C00 - 0x4000 6FFF	Backup registers (BKP)		Section 5.4.5 on page 69
0x4000 5C00 - 0x4000 6BFF	Reserved		-
0x4000 5800 - 0x4000 5BFF	I2C2		Section 22.6.10 on page 598
0x4000 5400 - 0x4000 57FF	I2C1		Section 22.6.10 on page 598
0x4000 5000 - 0x4000 53FF	UART5		Section 23.6.8 on page 646
0x4000 4C00 - 0x4000 4FFF	UART4		Section 23.6.8 on page 646
0x4000 4800 - 0x4000 4BFF	USART3		Section 23.6.8 on page 646
0x4000 4400 - 0x4000 47FF	USART2		Section 23.6.8 on page 646
0x4000 4000 - 0x4000 43FF	Reserved		-
0x4000 3C00 - 0x4000 3FFF	SPI3		Section 21.4.8 on page 565
0x4000 3800 - 0x4000 3BFF	SPI2		Section 21.4.8 on page 565
0x4000 3400 - 0x4000 37FF	Reserved		-
0x4000 3000 - 0x4000 33FF	Independent watchdog (IWDG)		Section 18.4.5 on page 486
0x4000 2C00 - 0x4000 2FFF	Window watchdog (WWDG)		Section 18.4.5 on page 486
0x4000 2800 - 0x4000 2BFF	RTC		Section 17.4.7 on page 480
0x4000 2400 - 0x4000 27FF	Reserved		-
0x4000 2000 - 0x4000 23FF	TIM14 timer		Section 14.5.11 on page 387
0x4000 1C00 - 0x4000 1FFF	TIM13 timer		Section 14.5.11 on page 387
0x4000 1800 - 0x4000 1BFF	TIM12 timer		Section 14.4.14 on page 377
0x4000 1400 - 0x4000 17FF	TIM7 timer		Section 16.4.9 on page 468
0x4000 1000 - 0x4000 13FF	TIM6 timer		Section 16.4.9 on page 468
0x4000 0C00 - 0x4000 0FFF	TIM5 timer		Section 13.4.19 on page 340
0x4000 0800 - 0x4000 0BFF	TIM4 timer		Section 13.4.19 on page 340
0x4000 0400 - 0x4000 07FF	TIM3 timer		Section 13.4.19 on page 340
0x4000 0000 - 0x4000 03FF	TIM2 timer		Section 13.4.19 on page 340

2.3.1 Embedded SRAM

The STM32F100xx features up to 32 Kbytes of static SRAM. It can be accessed as bytes, half-words (16 bits) or full words (32 bits). The SRAM start address is 0x2000 0000.

2.3.2 Bit banding

The Cortex[®]-M3 memory map includes two bit-band regions. These regions map each word in an alias region of memory to a bit in a bit-band region of memory. Writing to a word in the alias region has the same effect as a read-modify-write operation on the targeted bit in the bit-band region.

In the STM32F100xx, both peripheral registers and SRAM are mapped in a bit-band region. This allows single bit-band write and read operations to be performed.

A mapping formula shows how to reference each word in the alias region to a corresponding bit in the bit-band region. The mapping formula is:

$$bit_word_addr = bit_band_base + (byte_offset \times 32) + (bit_number \times 4)$$

where:

bit_word_addr is the address of the word in the alias memory region that maps to the targeted bit

bit_band_base is the starting address of the alias region

byte_offset is the number of the byte in the bit-band region that contains the targeted bit

bit_number is the bit position (0-7) of the targeted bit

Example:

The following example shows how to map bit 2 of the byte located at SRAM address 0x2000 0300 in the alias region:

$$0x2200\ 6008 = 0x2200\ 0000 + (0x300 \times 32) + (2 \times 4).$$

Writing to address 0x2200 6008 has the same effect as a read-modify-write operation on bit 2 of the byte at SRAM address 0x2000 0300.

Reading address 0x2200 6008 returns the value (0x01 or 0x00) of bit 2 of the byte at SRAM address 0x2000 0300 (0x01: bit set; 0x00: bit cleared).

For more information on bit-banding, refer to the *Cortex[®]-M3 Technical Reference Manual*.

2.3.3 Embedded flash memory

The high-performance flash memory module has the following key features:

- Density of up to 512 Kbytes
- Memory organization: the flash memory is organized as a main block and an information block:
 - Main memory block of size:
 - up to 8 Kbit × 32 bits divided into 32 pages of 1 Kbyte each for low-density value line devices (see [Table 3](#))
 - up to 32 Kbit × 32 bits divided into 128 pages of 1 Kbyte each for medium-density value line devices (see [Table 4](#))
 - up to 128 Kbit × 32 bits divided into 256 pages of 2 Kbyte each for high-density value line devices (see [Table 5](#))
 - Information block of size:
 - 516 × 32 bits for low, medium and high-density value line devices (see [Table 3](#), [Table 4](#) and [Table 5](#))

The flash memory interface (FLASH) features:

- Read interface (32-bit)
- Option byte loader
- Flash Program/Erase operation
- Read/write protection

Table 3. Flash module organization (low-density value line devices)

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	.	.	.
	Page 31	0x0800 7C00 - 0x0800 8000	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

Table 4. Flash module organization (medium-density value line devices)

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 03FF	1 Kbyte
	Page 1	0x0800 0400 - 0x0800 07FF	1 Kbyte
	Page 2	0x0800 0800 - 0x0800 0BFF	1 Kbyte
	Page 3	0x0800 0C00 - 0x0800 0FFF	1 Kbyte
	Page 4	0x0800 1000 - 0x0800 13FF	1 Kbyte
	.	.	.
	Page 127	0x0801 FC00 - 0x0801 FFFF	1 Kbyte
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRP	0x4002 2020 - 0x4002 2023	4

Table 5. Flash module organization (high-density value line devices)

Block	Name	Base addresses	Size (bytes)
Main memory	Page 0	0x0800 0000 - 0x0800 07FF	2 Kbytes
	Page 1	0x0800 0800 - 0x0800 0FFF	2 Kbytes
	Page 2	0x0800 1000 - 0x0800 17FF	2 Kbytes
	Page 3	0x0800 1800 - 0x0800 1FFF	2 Kbytes
	.	.	.
	Page 255	0x0807 F800 - 0x0807 FFFF	2 Kbytes
Information block	System memory	0x1FFF F000 - 0x1FFF F7FF	2 Kbytes
	Option Bytes	0x1FFF F800 - 0x1FFF F80F	16

Table 5. Flash module organization (high-density value line devices) (continued)

Block	Name	Base addresses	Size (bytes)
Flash memory interface registers	FLASH_ACR	0x4002 2000 - 0x4002 2003	4
	FLASH_KEYR	0x4002 2004 - 0x4002 2007	4
	FLASH_OPTKEYR	0x4002 2008 - 0x4002 200B	4
	FLASH_SR	0x4002 200C - 0x4002 200F	4
	FLASH_CR	0x4002 2010 - 0x4002 2013	4
	FLASH_AR	0x4002 2014 - 0x4002 2017	4
	Reserved	0x4002 2018 - 0x4002 201B	4
	FLASH_OBR	0x4002 201C - 0x4002 201F	4
	FLASH_WRPR	0x4002 2020 - 0x4002 2023	4

Note: For further information on the flash memory interface registers, refer to PM0063.

Reading flash memory

Flash memory accesses are performed through the AHB bus. Accesses are either instruction fetches over the ICode bus, or data accesses (e.g. literal pool) over the D-code bus. Since these two buses have the same flash memory as target, the interface gives priority to D-code bus accesses over I-code bus, instruction fetch accesses.

Read accesses can be performed without any wait state and with the following configuration options:

- Half cycle: for power optimization

- Note:**
- 1 *Half cycle configuration is not available in combination with a prescaler on the AHB. The system clock (SYSCLK) should be equal to the HCLK clock. This feature can therefore be used only with a low-frequency clock of 8 MHz or less. It can be generated from the HSI or the HSE but not from the PLL.*
 - 2 *Using DMA: DMA accesses flash memory on the DCode bus and has priority over ICode instructions. The DMA provides one free cycle after each transfer. Some instructions can be performed together with DMA transfer.*

Programming and erasing flash memory

The flash memory can be programmed 16 bits (half words) at a time. For write and erase operations on the flash memory (write/erase), the internal RC oscillator (HSI) must be ON.

The flash memory erase operation can be performed at page level or on the whole area (mass erase). Mass erase does not affect the information blocks.

To ensure that there is no overprogramming, the flash programming and erase controller blocks are clocked by a fixed clock.

The end of write operation (programming or erasing) can trigger an interrupt. This interrupt can be used to exit the WFI mode, only if the FLASH clock is enabled. Otherwise, the interrupt is served only after exiting WFI.

The FLASH_ACR register is used to enable/disable flash memory half-cycle access. The tables below provide the bit map and bit descriptions for this register.

For complete information on flash memory operations and register configurations, refer to PM00063).

Flash access control register (FLASH_ACR)

Address offset: 0x00

Reset value: 0x0000 0000

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

Bits 31:4 Reserved, must be kept cleared.

Bit 3 **HLFCYA**: Flash half cycle access enable

0: Half cycle is disabled

1: Half cycle is enabled

Bits 2:0 Reserved, must be kept cleared.

2.4 Boot configuration

In the STM32F100xx, three different boot modes can be selected through the BOOT[1:0] pins as shown in [Table 6](#).

Table 6. Boot modes

Boot mode selection pins		Boot mode	Aliasing
BOOT1	BOOT0		
x	0	Main flash memory	Main flash memory is selected as the boot space
0	1	System memory	System memory is selected as the boot space
1	1	Embedded SRAM	Embedded SRAM is selected as the boot space

The values on the BOOT pins are latched on the 4th rising edge of SYSCLK after a reset. It is up to the application to set the BOOT1 and BOOT0 pins after reset to select the required boot mode.

The BOOT pins are also resampled when exiting the Standby mode. Consequently, they must be kept in the required boot mode configuration in the Standby mode. After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code execution from the boot memory starting from 0x0000 0004.

Due to its fixed memory map, the code area starts from address 0x0000 0000 (accessed through the ICode/DCode buses) while the data area (SRAM) starts from address 0x2000 0000 (accessed through the system bus). The Cortex-M3 CPU always fetches the reset vector on the ICode bus, which implies to have the boot space available only in the code area (typically, flash memory). STM32F100xx microcontrollers implement a special mechanism to be able to boot also from SRAM and not only from main flash memory and System memory.

Depending on the boot mode selected, the main flash memory, system memory or SRAM is accessible as follows:

- Boot from main flash memory: the main flash memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x800 0000). In other words, the flash memory contents can be accessed starting from address 0x0000 0000 or 0x800 0000.
- Boot from system memory: the system memory is aliased in the boot memory space (0x0000 0000), but still accessible from its original memory space (0x1FFF F000).
- Boot from the embedded SRAM: SRAM is accessible only at address 0x2000 0000.

Note: When booting from SRAM, in the application initialization code, the vector table in SRAM must be relocated using the NVIC exception table and offset register.

Embedded boot loader

The embedded boot loader is used to reprogram the flash memory using the USART1 serial interface. This program is located in the system memory and is programmed by ST during production. For further details refer to AN2606.