PY32F030 series
ARM® 32-bit Cortex®-M0+ microcontrollers

Reference manual

Contents

1. List of abbreviations for register

Abbreviation	Describe
Read/write (rw)	Software can read and write to this bit.
Read-only (r)	Software can only read this bit.
Write-only (w)	Software can only write to this bit. Reading this bit returns the reset value.
Read/clear write0 (rc_w0)	Software can read as well as clear this bit by writing 0. Writing ‘1’ has no effect on the bit value.
Read/clear write1 (rc_w1)	Software can read as well as clear this bit by writing 1. Writing ‘0’ has no effect on the bit value.
Read/clear write (rc_w)	Software can read as well as clear this bit by writing register. Writing to this bit has no effect.
Read/clear by read (rc_r)	Software can read this bit. Reading this bit automatically clears it to ‘0’. Writing this bit has no effect on the bit value.
Read/set by read (rs_r)	Software can read this bit. Reading this bit automatically clears it to ‘0’. Writing this bit has no effect on the bit value.
Read/set (rs)	Software can read as well as set this bit to ‘1’. Writing ‘0’ has no effect on the bit value.
Toggle (t)	Software can toggle this bit by writing ‘1’. Writing ‘0’ has no effect.
Reserved (Res)	Reserved bit, must be kept at reset value.

2. System architecture

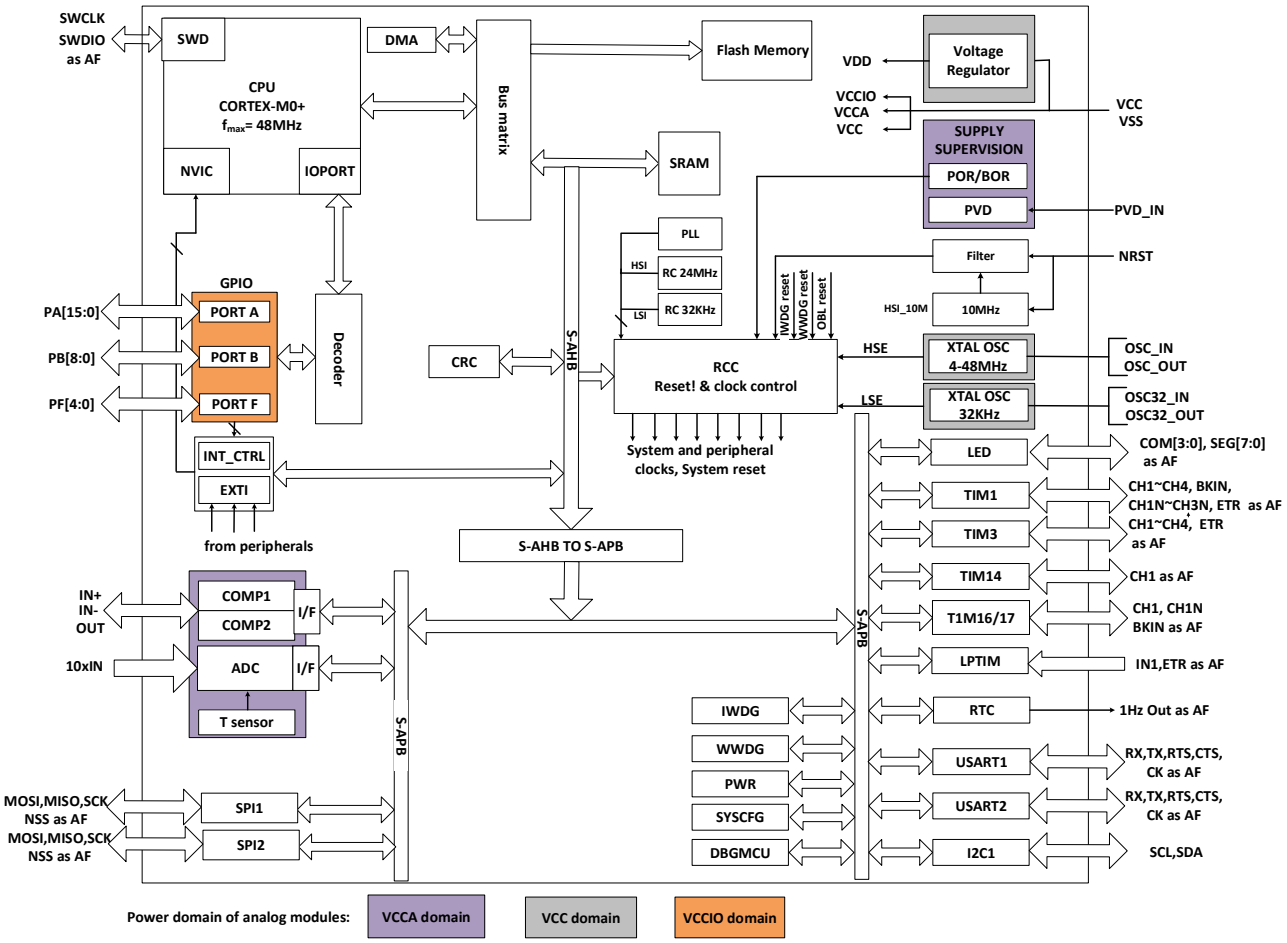


Figure 2-1 System architecture

3. Memory and bus architecture

3.1. System architecture

The system consists of the following parts:

- Two masters:
  - Cortex-M0+
  - General-purpose DMA
- Three Slaves
  - Internal SRAM
  - Internal Flash memory
  - AHB with AHB-APB Bridge

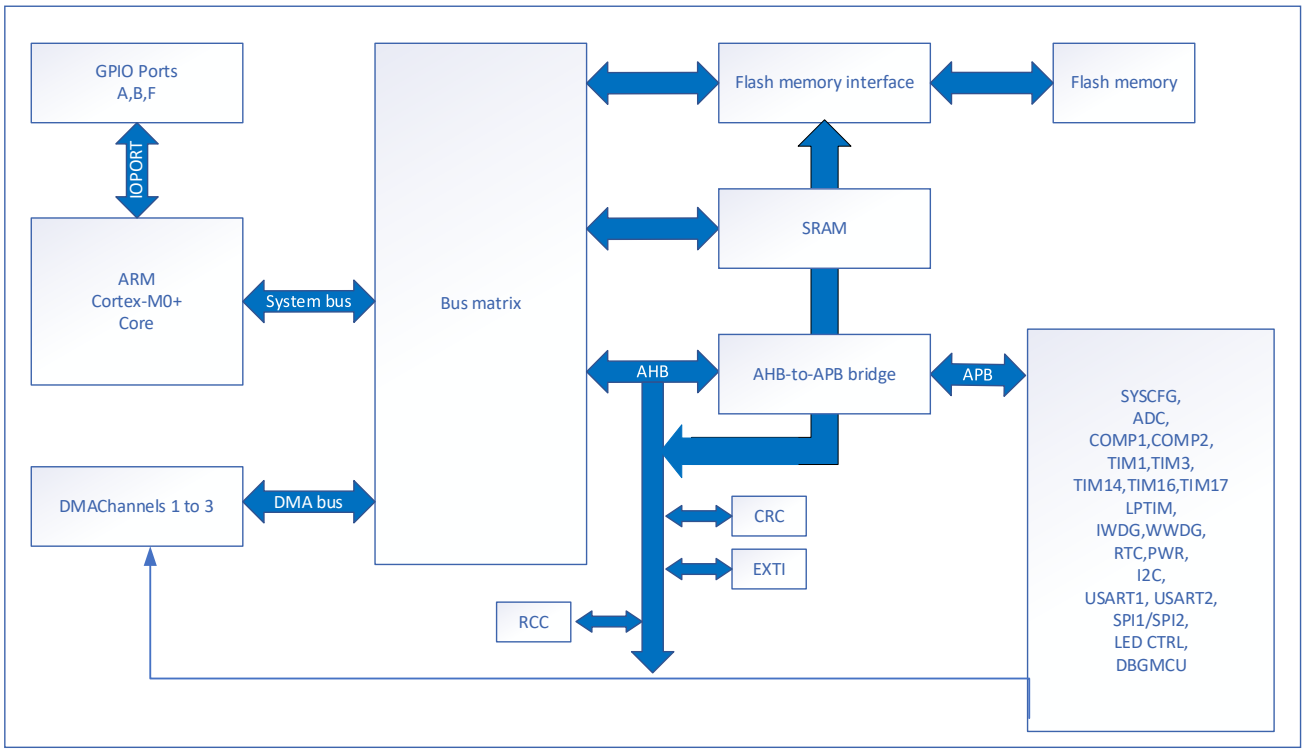


Figure 3-1 System architecture

- System bus

This bus connects the system bus of the Cortex-M0+ core to a BusMatrix which manages the arbitration between the CPU and the DMA.

- DMA bus

The bus connects the AHB master interface of the DMA to the BusMatrix which manages the access of CPU and DMA to SRAM, Flash memory and peripherals of AHB/APB.

- BusMatrix

The BusMatrix manages the access arbitration between the CPU bus and the DMA master bus. This arbitration uses the Round Robin algorithm. The BusMatrix is composed of masters (CPU and DMA) and slaves (SRAM, Flash memory and AHB-to-APB bridge).

- AHB-to-APB bridge (APB)

The AHB-to-APB bridge provides a synchronous connection between the AHB and APB buses to the peripheral address mapping of the Bridge.

3.2. Memory organization

3.3. Introduction to memory structure

Program memory, data memory, registers and I/O ports are organized within the same linear 4-Gbytes address space. The bytes are coded in memory in Little Endian format (in a word, the lowest numbered byte is considered the world's least sinificant byte).

The addressable memory space is divided into 8 mian blocks, each of 512 Mbyte.

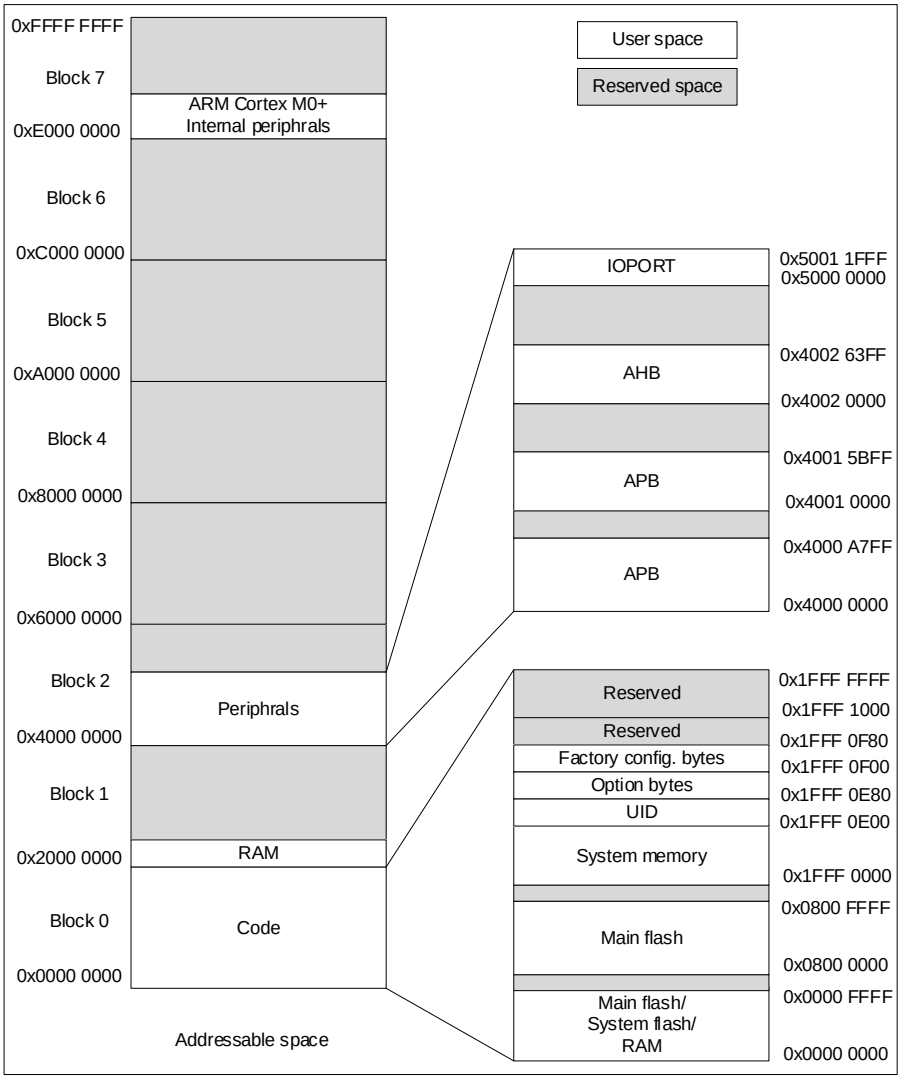


Figure 3-2 Memory map

Table 3-1 Memory boundary addresses

Type	Boundary Address	Size	Memory Area	Description
SRAM	0x2000 2000 - 0x3FFF FFFF	512 Mbytes	Reserved	
	0x2000 0000 - 0x2000 1FFF	8 kbytes	SRAM	Depending on the hardware, the maximum SRAM is 8 kbytes
Code	0x1FFF 1000 - 0x1FFF FFFF	4 kbytes	Reserved	
	0x1FFF 0F80 - 0x1FFF 0FFF	128 bytes	Reserved	

	0x1FFF 0F00 - 0x1FFF 0F7F	128 bytes	Factory config	Store HSI triming data, Flash erasing time configuration parameters
	0x1FFF 0E80 - 0x1FFF 0EFF	128 bytes	Option bytes	option bytes
	0x1FFF 0E00 - 0x1FFF 0E7F	128 bytes	UID	Unique ID
	0x1FFF 0000 - 0x1FFF 0DFF	3.5 kbytes	System memory	Store the boot loader
	0x0801 0000 - 0x1FFF FFFF	384 Mbytes	Reserved	
	0x0800 0000 - 0x0800 FFFF	64 kbytes	Main Flash memory	
	0x0001 0000 - 0x07FF FFFF	8 Mbytes	Reserved	
	0x0000 0000 - 0x0000 FFFF	64 kbytes	According to the Boot configuration: 1) Main Flash memory 2) System memory 3) SRAM	

Note:

Except for 0x1FFF 0E00 - 0x1FFF 0E7F, the above spaces are marked as reserved spaces, which cannot be written and read as 0 with a response error occurs.

Table 3-2 Peripheral register address

Bus	Boundary Address	Size	Peripheral
	0xE000 0000 - 0xE00F FFFF	1 Mbytes	M0+
I/O PORT	0x5000 1800 - 0x5FFF FFFF	256 Mbytes	Reserved (1)
	0x5000 1400 - 0x5000 17FF	1 kbytes	GPIOF
	0x5000 1000 - 0x5000 13FF	1 kbytes	Reserved
	0x5000 0C00 - 0x5000 0FFF	1 kbytes	Reserved
	0x5000 0800 - 0x5000 0BFF	1 kbytes	Reserved
	0x5000 0400 - 0x5000 07FF	1 kbytes	GPIOB
	0x5000 0000 - 0x5000 03FF	1 kbytes	GPIOA
AHB	0x4002 3400 - 0x4FFF FFFF		Reserved
	0x4002 300C - 0x4002 33FF	1 kbytes	Reserved
	0x4002 3000 - 0x4002 3008		CRC
	0x4002 2400 - 0x4002 2FFF		Reserved
	0x4002 2124 - 0x4002 23FF	1 kbytes	Reserved
	0x4002 2000 - 0x4002 2120		Flash
	0x4002 1C00 - 0x4002 1FFF	3 kbytes	Reserved
	0x4002 1888 - 0x4002 1BFF	1 kbytes	Reserved
	0x4002 1800 - 0x4002 1884		EXTI (2)
	0x4002 1400 - 0x4002 17FF	1 kbytes	Reserved
	0x4002 1064 - 0x4002 13FF	1 kbytes	Reserved
	0x4002 1000 - 0x4002 1060		RCC (2)
	0x4002 0C00 - 0x4002 0FFF	1 kbytes	Reserved
	0x4002 0040 - 0x4002 03FF	1 kbytes	Reserved
	0x4002 0000 - 0x4002 003C		DMA
APB	0x4001 5C00 - 0x4001 FFFF	32 kbytes	Reserved
	0x4001 5880 - 0x4001 5BFF	1 kbytes	Reserved
	0x4001 5800 - 0x4001 587F		DBG
	0x4001 4C00 - 0x4001 57FF	3 kbytes	Reserved
	0x4001 4850 - 0x4001 4BFF	1 kbytes	Reserved
	0x4001 4800 - 0x4001 484C		TIM17
	0x4001 4450 - 0x4001 47FF	1 kbytes	Reserved
	0x4001 4400 - 0x4001 404C		TIM16
	0x4001 3C00 - 0x4001 43FF	2 kbytes	Reserved
	0x4001 381C - 0x4001 3BFF	1 kbytes	Reserved
	0x4001 3800 - 0x4001 3018		USART1
	0x4001 3400 - 0x4001 37FF	1 kbytes	Reserved

	0x4001 3010 - 0x4001 33FF	1 kbytes	Reserved
	0x4001 3000 - 0x4001 300C		SPI1
	0x4001 2C50 - 0x4001 2FFF	1 kbytes	Reserved
	0x4001 2C00 - 0x4001 2C4C		TIM1
	0x4001 2800 - 0x4001 2BFF	1 kbytes	Reserved
	0x4001 270C - 0x4001 27FF	1 kbytes	Reserved
	0x4001 2400 - 0x4001 2708		ADC
	0x4001 0400 - 0x4001 23FF	8 kbytes	Reserved
	0x4001 0220 - 0x4001 03FF	1 kbytes	Reserved
	0x4001 0200 - 0x4001 021F		COMP1 and COMP2
	0x4001 0000 - 0x4001 01FF		SYSCFG
	0x4000 B400 - 0x4000 FFFF	19 kbytes	Reserved
	0x4000 B000 - 0x4000 B3FF	1 kbytes	Reserved
	0x4000 8400 - 0x4000 AFFF	11 kbytes	Reserved
	0x4000 8000 - 0x4000 83FF	1 kbytes	Reserved
	0x4000 7C28 - 0x4000 7FFF	1 kbytes	Reserved
	0x4000 7C00 - 0x4000 7C24		LPTIM
	0x4000 7400 - 0x4000 7BFF	2 kbytes	Reserved
	0x4000 7018 - 0x4000 73FF	1 kbytes	Reserved
	0x4000 7000 - 0x4000 7014		PWR (3)
	0x4000 5800 - 0x4000 6FFF	6 kbytes	Reserved
	0x4000 5434 - 0x4000 57FF	1 kbytes	Reserved
	0x4000 5400 - 0x4000 5430		I2C
	0x4000 4800 - 0x4000 53FF	3 kbytes	Reserved
	0x4000 441C - 0x4000 47FF	1 kbytes	Reserved
	0x4000 4400 - 0x4000 4418		USART2
	0x4000 3C00 - 0x4000 43FF	1 kbytes	Reserved
	0x4000 3810 - 0x4000 3BFF	1 kbytes	Reserved
	0x4000 3800 - 0x4000 380C		SPI2
	0x4000 3400 - 0x4000 37FF	1 kbytes	Reserved
	0x4000 3014 - 0x4000 33FF	1 kbytes	Reserved
	0x4000 3000 - 0x4000 0010		IWDG
	0x4000 2C0C - 0x4000 2FFF	1 kbytes	Reserved
	0x4000 2C00 - 0x4000 2C08		WWDG
	0x4000 2830 - 0x4000 2BFF	1 kbytes	Reserved
	0x4000 2800 - 0x4000 282C		RTC (3)
	0x4000 2420 - 0x4000 27FF	1 kbytes	Reserved
	0x4000 2400 - 0x4000 241C		LED
	0x4000 2054 - 0x4000 23FF	1 kbytes	Reserved
	0x4000 2000 - 0x4000 0050		TIM14
	0x4000 1800 - 0x4000 1FFF	2 kbytes	Reserved
	0x4000 1400 - 0x4000 17FF	1 kbytes	Reserved
	0x4000 1030 - 0x4000 13FF	1 kbytes	Reserved
	0x4000 1000 - 0x4000 102C		Reserved
	0x4000 0800 - 0x4000 0FFF	2 kbytes	Reserved
	0x4000 0450 - 0x4000 07FF	1 kbytes	Reserved
	0x4000 0400 - 0x4000 044C		TIM3
	0x4000 0000 - 0x4000 03FF	1 kbytes	Reserved

Note :

(1) In the above table, the address space which marked as Reserved in AHB, cannot be written, read is 0 and a hardfault is generated. And in APB is marked as the reserved address space, cannot be written, read is 0, and no hardfault will be generated.

(2) Not only supports 32 bit word access, but also supports halfword and byte access.

(3) Not only supports 32 bit word access, but also supports half word access.

3.4. Embedded SRAM

The PY32F030 features up to 8 kbytes of SRAM. It can be accessed as bytes, half-word (16 bits) or full words (32 bits). A hard fault will be generated when the software reads and writes the space outside the setting range.

3.5. Flash memory

Flash memory consists of two physical areas:

- Main Flash area, 64 kbytes, it contains application and user data.
- Information area, 4 kbytes, it includes the following parts:
  - Factory config. bytes: 128 bytes, used to store trimming data (including HSI triming data), power-on reading check code, etc.
  - UID: 128 bytes, used to store the UID of the chip
  - Option bytes: 128 bytes, used to store the configuration values of hardware and storage protection
  - System memory (system memory): 3.5 kbytes, used to store Boot loader

Flash memory interface implements instruction of reading and data access based on the AHB protocol, and it also implements the basic program/erase operations of the Flash through registers.

3.6. Boot mode

Three different boot mode can be selected through the BOOT0 pin and boot selector option bit nBOOT1 (stored in the Option bytes), as shown in the following table:

Table 3-3 Boot mood

Boot mode selector pins		Mode
nBOOT1 bit	BOOT0 pin	
X	0	Main Flash memory is selected as the boot area
1	1	System memory is seleced as the boot area
0	1	Embedded SRAM is selected as the boot area

The values on the Boot pins are latched on the 4th SYSCLK after a reset. It is up to the user to set the boot mode to choose according to the table above.

After this startup delay has elapsed, the CPU fetches the top-of-stack value from address 0x0000 0000, then starts code executes from the boot memory starting from 0x0000 0004. Depending on the selected boot mode, 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 (0x0800 0000). In other word, the Flash memory contents can be accessed starting from address 0x0000 0000 or 0x0800 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 0000).
- Boot from the embedded SRAM: the SRAM is aliased in the boot memory space (0x0000 0000), but still accessible at address 0x2000 0000.

3.6.1. Memory physical mapping

If boot mode is selected, the application software can modify the memory accessible in the program space. This modification is determined by the MEM_MODE bit selection in the SYSCFG_CFGR1 register (see the SYSCFG chapter for details).

3.6.2. Embedded boot loader

The embedded boot loader is located in the System memory, programmed during production. It is used to re-program the Flash memory using the following serial interface:

- USART corresponding to PA14/PA15 or PA9/PA10 or PA2/PA3.

4. Embedded Flash memory

4.1. Key features

- Main Flash block: maximum 64 kbytes (16 k x 32 bit)
- Information block: 4 kbytes (1 k x 32 bit)
- Page size: 128 bytes
- Sector size: 4 kbytes

The Flash control interface circuit features:

- Flash write and erase
- Read protection
- Write protection

4.2. Flash memory function introduction

4.2.1. Flash structure

Flash memory is composed of 32-bit wide storage units, which can be used for program and data storage.

In terms of function, Flash memory is divided into main Flash and information Flash, the former has a maximum capacity of 64 kbytes, and the latter has a capacity of 4 kbytes.

Page erase operation can be applied to main Flash and system Flash, but sector erase can not be applied to system Flash.

Mass erase can be applied to main Flash if there is no write protection setting, otheRWise it cannot be applied to main Flash. Mass erase cannot be applied to system Flash with or without write protection.

Table 4-1 Flash structure and boundary addresses

Block	Sector	Page	Base address	Size
Main Flash	Sector 0	Page 0-31	0x0800 0000 - 0x0800 0FFF	4 kbytes
	Sector 1	Page 32-63	0x0800 1000 - 0x0800 1FFF	4 kbytes
	Sector 2	Page 64-95	0x0800 2000 - 0x0800 2 FFF	4 kbytes
	…	…	…	…
	Sector 14	Page 448-479	0x0800 E000 - 0x0800 EFFF	4 kbytes
	Sector 15	Page 480-511	0x0800F000 - 0x0800FFFF	4 kbytes
System Flash	Sector 16	Page 0-27	0x1FFF 0000 - 0x1FFF 0DFF	3.5 kbytes
UID		Page 28	0x1FFF 0E00 - 0x1FFF 0E7F	128 bytes
Option bytes		Page 29	0x1FFF 0E80 - 0x1FFF 0EFF	128 bytes
Factory config		Page 30	0x1FFF 0F00 - 0x1FFF 0F7F	128 bytes
Reserved		Page 31	0x1FFF 0F80 - 0x1FFF 0FFF	128 bytes

4.2.2. Flash read operation and access latency

Flash can be used as a general memory space to accessed direct addressing. The contents of the Flash memory can be read through a special read control sequence.

The instruction fetch and data access are both done through the AHB bus. Read can mange through the Latency of the FLASH_ACR register, which is the read operation increase the wait state or not. When it is 0, the wait state of the Flash read operation is not added, when it is 1, the Flash read operation adds one wait state. This mechanism is specially designed to match high-speed system clock and relatively low-speed Flash read speed.

4.2.3. Flash program and erase operations

The Flash memory can be programmed by In -circuit programming (ICP) or In -application programming (IAP).

ICP: It is used to update the entire contents of the Flash memory, using the SWD protocol or the boot loader to load the user application into the MCU. ICP provides quick and efficient design iterations and eliminates unnecessary package handling or socketing of devices.

IAP: It can use any communication interface supported by the microcontroller to download programming data into Flash memory. The IAP allows the user to re-program the Flash memory while the application is running. Then, part of the application has to have been previously programmed in the Flash memory using ICP.

If a reset occurs during Flash program and erase operations, the contents of the Flash memory are not protected.

During a program and erase operations to the Flash memory, any attempt to read the Flash memory will stall the bus. The read operation will proceed correctly once the program and erase operations has completed. This means that code or data fetches cannot be made while programming and erasing operations are in progress.

For program and erase operations, the HSI must be turned on.

Program and erase operations can be implemented through the following control interface-related registers :

- Access control register (FLASH_ACR)
- KEY register (FLASH_KEYR)
- Option byte key register (FLASH_OPTKEYR)
- Flash status register (FLASH_SR)
- Flash control register (FLASH_CR)
- Flash option register (FLASH_OPTR)
- Flash special area address register (FLASH_SAR)
- Flash w rite protection resister (FLASH_WRPR)
- Flash TS0 register (FLASH_TS0)
- Flash TS1 register (FLASH_TS1)
- TS2P register (FLASH_TS2P)
- Flash TPS3 register (FLASH_TPS3)
- Flash TS3 register (FLASH_TS3)
- Flash page erase TPE register (FLASH_PERTPE)
- Flash sector/mass erase TPE register (FLASH_SMERTPE)
- Flash program TPE register (FLASH_PRGTPE)
- Flash pre-program TPE register (FLASH_PRETPE)

4.2.3.1. Unlocking the Flash memory

After reset, the Flash memory is protected against unwanted (like caused by electrical interference) write or erase operations. The FLASH_CR register is not accessible in write mode, except for the OB L_LAUNCH bits, used to reload option bit. Every time to write or erase the Flash, must write the FLASH_KEYR register, to generate an unlock sequence, and to open the access to the FLASH_CR register.

This sequence consists of two steps:

Step 1: Write KEY1 = 0x4567 0123 to the FLASH_KEYR register

Step 2: Write KEY2 = 0xCDEF 89AB to the FLASH_KEYR register

Any wrong sequence locks up the FLASH_CR register until the next reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated. This is done after the first write cycle if KEY1 does not match, or during the second write cycle if KEY1 has been correctly written but KEY2 does not match.
The FLASH_CR register can be locked again by user software by writing the LOCK bit in the FLASH_CR register.
In addition, the FLASH_CR register cannot be written when the BSY bit of the FLASH_SR register is set. In the meantime, any attempt to write FLASH_CR register will cause the AHB bus to stall until the BSY1 bit is cleared.

4.2.3.2. Flash memory programming

The Flash memory can be programmed the entire page in units of 32 bits each time (hardfault will be generated when the half word or byte operation is performed). The program operation is started when the CPU writes a half-word into a main Flash memory address with the PG bit of the FLASH_CR register set. Any non 32-bit write will cause a hard fault interrupt.
If the address is write-protected by the FLASH_WRPR register, the program operation is skipped and a warning is issued by the WRPRTERR bit in the FLASH_CR register. At the end of the program operation, the EOP bit in the FLASH_CR register will be set.
The Flash memory programming sequence is as follows:

1) Check that no Flash memory operation is ongoing by checking the BSY in the FLASH_SR register.
2) If no Flash memory erase or program operation is ongoing, the software reads out the 32 words of the page (if the page already has data stored, perform this step, otheRWise skip this step).
3) To release the protection of the FLASH_CR register by programming KEY1 and KEY2 to the FLASH_KEYR register.
4) Set the PG bit and the EOPIE bit in the FLASH_CR register.
5) Programming to the target address from the 1st to 31st word (only accept 32 bit program).
6) Set the PGSTRT in FLASH_CR register.
7) Write the 32nd word.
8) Wait until the BSY bit of the FLASH_SR register to be cleared.
9) Check the EOP flag in the FLASH_SR register (It is set when the programming operation has succeeded), and then clear it by software.
10) If there are no more program operations, software will clear the PG bit.

When the above step 7) is successfully executed, the program operation is automatically started, and the BSY bit is set by hardware at the same time.

Flash Erase Operation

The Flash memory can be erased by page, or sector and mass erase (sector and mass erase do not work for information memory).

4.2.3.3. Page erase

When a page is protected by WRP, it will not be erased and the WRPERR bit is set at this time. To excution the page erase operation, the following steps need to be performed:

1) Check that no Flash memory operation is ongoing by checking the BSY in the FLASH_SR register.
2) To release the protection of the FLASH_CR register by programming KEY1 and KEY2 to the FLASH_KEYR register.
3) Set the PER bit and the EOPIE bit in the FLASH_CR register.

4) Write arbitrary data (32-bit data) to the page.
5) Wait for the BSY bit to be cleared.
6) Check that the EOP flag is set.
7) Clear the EOP flag.

4.2.3.4. Mass erase

The Mass erase can used to completely erase the entire main Flash memory, but the information block is unaffected by this procedure. Additionally, when WRP is enabled, the mass erase function is disabled and no mass erase operation occurs, the WEPERR bit is set.

The following sequence for mass erase:

1) Check that no Flash memory operation is ongoing by checking the BSY.
2) To release the protection of the FLASH_CR register by programming KEY1 and KEY2 to the FLASH_KEYR register.
3) Set the MER bit and the EOPIE bit in the FLASH_CR register.
4) Write arbitrary data (32-bit data) to the main Flash memory.
5) Wait for the BSY bit to be cleared.
6) Check that the EOP flag is set.
7) Clear the EOP flag.

4.2.3.5. Sector erase

The wector erase can be used to erase the main Flash of 4 kbytes, but the information block is unaffected by this procedure. In addition, when a sector is protected by WRP, it will not be erased, and the WRPERR bit is set.

The following sequence for sector erase:

1) Check that no Flash memory operation is ongoing by checking the BSY.
2) To release the protection of the FLASH_CR register by programming KEY1 and KEY2 to the FLASH_KEYR register.
3) Set the SER bit and the EOPIE bit in the FLASH_CR register.
4) Write arbitrary data to the sector.
5) Wait for the BSY bit to be cleared.
6) Check that the EOP flag is set.
7) Clear the EOP flag.

System memory is read-only and will never be programmed/erased.

4.2.3.6. Program and erase time configuration

The program and erase time need to be strictly controlled, otheRWise the operation will fail. By default, the hardware design sets the time parameters of program and erase operations that the HSI is 24 MHz. When the HSI output frequency is changed, the Flash program and erase time need to be configured the register correctly according to the table below.

Table 4-2 Program and erase time configuration

Register	4 MHz	8 MHz	16 MHz	22.12 MHz	24 MHz
TS0	0x1E	0x3C	0x78	0xA6	0xB4
TS1	0x48	0x90	0x120	0x18F	0x1B0
TS2P	0x1E	0x3C	0x78	0xA6	0xB4
TPS3	0x120	0x240	0x480	0x639	0x6C0

TS3	0x1E	0x3C	0x78	0xA6	0xB4
PERTPE	0x2EE0	0x5DC0	0XBB80	0x10338	0x11940
SMERTPE	0x2EE0	0x5DC0	0XBB80	0x10338	0x11940
PRGTPE	0XFA0	0x1F40	0x3E80	0x5668	0x5DC0
PRETPE	0x320	0x640	0xC80	0x1148	0x12C0

4.3. Flash option byte

4.3.1. Flash option word

Part of the information area is used as an option byte, which is used to store the hardware configuration that the chip or the user needs to perform for the application. For example, the watchdog can be selected in hardware or software mode.

For data security, the option bytes are stored separately in the code and one's complement code.

Table 4-3 Option byte format

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

The option bytes can be read from the memory locations listed in the table option byte organization or from the relevant registers of the following option bytes :

- FLASH user option register (FLASH_OPTR)
- FLASH SDK area address register (FLASH_SDKR)
- FLASH WRP address register (FLASH_WRPR)

Table 4-4 Option byte organization

Word address	Describe
0x1FFF 0E80	Option byte for Flash User option and its complemented
0x1FFF 0E84	Option byte for Flash SDK area address and its complemented
0x1FFF 0E88	Reserved
0x1FFF 0E8C	Option byte for Flash WRP address and its complemented
0x1FFF 0E90	Reserved
0x1FFF 0E94	Reserved
…	Reserved
…	Reserved
…	Reserved
0x1FFF 0EFC	Reserved

- **Option byte for Flash User option**

Flash memory Address offset: 0x1FFF 0E80

Production value: 0x0155 BEAA

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash information memory and written to the corresponding option bit of the register.

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
~nBOOT1	~NRST_ MODE	~WWDG _SW	~IWDG _SW	~BOR_LEV[2:0]			~BOR_ EN	~RDP[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
nBOOT1	NRST_ MODE	WWDG _SW	IWDG _SW	BOR_LEV[2:0]			BOR_ EN	RDP[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Function
31	~nBOOT1	R	One’s complement of nBOOT1
30	~NRST_MODE	R	One’s complement of NRST_MODE

29	~WWDG_SW	R	One's complement of WWDG_SW
28	~IWDG_SW	R	One's complement of IWDG_SW
27:25	~BOR_LEV[2:0]	R	One's complement of BOR_LEV
24	~BOR_EN	R	One's complement of BOR_EN
23:16	~RDP	R	One's complement of RDP
15	nBOOT1	R	Select boot mode with BOOT PIN
14	NRST_MODE	R	0: Reset input only 1: GPIO function
13	WWDG_SW	R	0: Hardware watchdog 1: Software watchdog
12	IWDG_SW	R	0: Hardware watchdog 1: Software watchdog
11:9	BOR_LEV[2:0]	R	000: BOR rising threshold is 1.8 V, falling threshold is 1.7 V 001: BOR rising threshold is 2.0 V, falling threshold is 1.9 V 010: BOR rising threshold is 2.2 V, falling threshold is 2.1 V 011: BOR rising threshold is 2.4 V, falling threshold is 2.3 V 100: BOR rising threshold is 2.6 V, falling threshold is 2.5 V 101: BOR rising threshold is 2.8 V, falling threshold is 2.7 V 110: BOR rising threshold is 3.0 V, falling threshold is 2.9 V 111: BOR rising threshold is 3.2 V, falling threshold is 3.1 V
8	BOR_EN	R	BOR enable 0: BOR is disabled 1: BOR is enabled, BOR_LEV works
7:0	RDP	R	0 xAA: level 0, read protection inactive Non 0xAA: level 1, read protection active

- Option byte for Flash SDK area address

Flash memory Address offset: 0x1FFF 0E84

Production value: 0x FF00 00FF

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash information memory and written to the corresponding option bit of the register.

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

Bit	Name	R/W	Function
31:16	Reserved		
28:24	Complemented SDK_END[4:0]	R	One's complement of SDK_END
23:21	Reserved		
20:16	Complemented SDK_STRT[4:0]	R	One's complement of SDK_STRT
15:13	Reserved		
12:8	SDK_END[4:0]	R	SDK area end address, each corresponding STEP is 2 kbytes
7:5	Reserved		
4:0	SDK_STRT[4:0]	R	SDK area start address, each corresponding STEP is 2 kbytes

- Option byte for Flash WRP address

Flash memory Address offset: 0x1FFF 0E8C

Production value: 0x0000 FFFF

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash information memory and written to the corresponding option bit of the register.

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

Bit	Name	R/W	Function
31:16	Complemented WRP	R	One's complement of WRP
15:0	WRP	R	0: sector [y] is protected 1: sector [y] unprotected y = 0 to 15

4.3.2. Flash option byte write

After reset, the bits in the FLASH_CR register associated with the option byte are write-protected. The OPTLOCK bit in the FLASH_CR register must be cleared before the option byte can be manipulated.

The following steps are used to unlock this register:

1) Unlock sequence to unlock write protection of FLASH_CR register.
2) Write OPTKEY1 = 0x0819 2A3B to the FLASH_OPTKEYR register.
3) Write OPTKEY2 = 0x4C5D 6E7F to the FLASH_OPTKEYR register.

Any wrong sequence locks up the FLASH_CR register until the next reset. In the case of a wrong key sequence, a bus error is detected and a Hard Fault interrupt is generated.

User option (option bytes in information Flash memory) can be protected by software by writing the OPTLOCK bit of the FLASH_CR register to prevent unwanted erase/program operations.

If software sets the Lock bit, the OPTLOCK bit is also automatically set.

Modifying user option bytes

Programming operation of the option byte is different from the operation to the main Flash memory. To modify the option bytes, the following steps are required:

1) Using the steps described previously to clear the OPTLOCK bit.
2) Check that no Flash memory operation is ongoing by checking the BSY.
3) Write the desired value (1~3 words) to the option bytes register FLASH_OPTR/ FLASH_SAR/ FLASH_WRPR.
4) Set OPTSTRT bit.
5) Write any 32 bit data to the main Flash memory address 0x4002 2080 (trigger a formal program operation).
6) Wait for the BSY bit to be cleared.
7) Wait for EOP to be pulled high, software to be cleared.

Any change to the option bytes, the hardware will first erase the entire page to the option byte, and then program the value of the FLASH_OPTR, FLASH_SAR or FLASH_WRPR register to the option bytes. And, the hardware automatically calculates the corresponding complement, and programs the calculated value to the corresponding area of the option bytes.

Option byte loading

After the BSY bit is cleared, all new option bytes are written into the Flash information memory, but they are not applied to the system. The read operation of the option bytes register still returns the value in the last loaded option bytes. Once they are loaded with new values, it will work on the system.

The loading of option bytes is performed in the following two cases:

- OBL_LAUNCH bit in the FLASH_CR register is set.
- After power-on reset (POR, BOR)

Loading option bytes is: read the option bytes in the information memory area, and then store the read data in the internal option registers (FLASH_OPTR, FLASH_SAR and FLASH_WRPR). These internal registers configure the system and can be read by software. The OBL_LAUNCH bit is set to generate a reset, so that the loading of option bytes can be carried out under the reset of the system.

Each option bit has a corresponding complement at its same doubleword address (next half word). During the loading of the option bytes, the validation of the option bit and its complement ensures that the loading was performed correctly.

If the one's complement matches, the option bytes are copied into the option register.

If the one's complement does not match, the OPTVERR status bit in the FLASH_SR register is set. Unmatched values are written to the option register:

- For user option
  - BOR_LEV is written as 000 (the lowest threshold)
  - The BOR_EN bit is written as 0 (BOR is not enabled)
  - NRST_MODE bit written to 0 (reset input only)
  - RDP bit is written as 0xff (which is level 1)
  - The rest of the mismatched values are written as 1
- For SDK area option, SDKR_STRT [4:0] = 0x00, SDKR_END [4:0] = 0x1F, all Flash memory is set as SDK
- For the WRP option, the unmatched value is the default "no protection"

After system reset, the contents of option bytes are copied to the following option registers (readable and writable by software):

- FLASH_OPTR
- FLASH_SDKR
- FLASH_WRPR

These registers are also used to modify option bytes. If these registers are not modified by the user, they reflect the state of the system option.

4.4. Flash configuration bytes

Part of the interval (one page in total) of the information area of the Flash memory is used as factory config. byte.

Page 0 is stored for software to read information (only code, no one's complement code is stored):

- HSI frequency selection control value, and corresponding trimming value.
- Erase and program time configuration parameter values corresponding to different frequencies of HSI.

Table 4-5Factory config. byte organization

Page	Word	Address	Contents
0	0	0x1FFF 0F00	Store HSI 4 MHz frequency selection control and corresponding trimming value
	1	0x1FFF 0F04	Store HSI 8 MHz frequency selection control and corresponding trimming value
	2	0x1FFF 0F08	Store HSI 16 MHz frequency selection control and corresponding trimming value
	3	0x1FFF 0F0C	Store HSI 22.12 MHz frequency selection control and corresponding trimming value
	4	0x1FFF 0F10	Store HSI 24 MHz frequency selection control and corresponding trimming value
	5	0x1FFF 0F14	TS_CAL1, 30℃ temperature sensor calibration value
	6	0x1FFF 0F18	TS_CAL2, 85℃ temperature sensor calibration value
	7	0x1FFF 0F1C	Store the configuration values of the corresponding FLASH_TSO and FLASH_TS1 registers at the HSI 4 MHz frequency

	8	0x1FFF 0F20	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at the HSI 4 MHz frequency
	9	0x1FFF 0F24	Store the configuration value of the corresponding FLASH_PERTPE register at the HSI 4 MHz frequency
	10	0x1FFF 0F28	Store the configuration value of the corresponding FLASH_ SMERTPE register at the HSI 4MHz frequency
	11	0x1FFF 0F2C	Store the configuration values of the corresponding FLASH_P RGTPE and FLASH_PRETPE registers at the HSI 4 MHz frequency
	12	0x1FFF 0F30	Store the configuration values of the corresponding FLASH_TSO and FLASH_TS1 registers at the HSI 8 MHz frequency
	13	0x1FFF 0F34	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at the HSI 8 MHz frequency
	14	0x1FFF 0F38	Store the configuration value of the corresponding FLASH_PERTPE register at the HSI 8 MHz frequency
	15	0x1FFF 0F3C	Store the configuration value of the corresponding FLASH_ SMERTPE register at the HSI 8 MHz frequency
	16	0x1FFF 0F40	Store the configuration values of the corresponding FLASH_PRGTPE and FLASH_PRETPE registers at the HSI 8 MHz frequency
	17	0x1FFF 0F44	Store the configuration values of the corresponding FLASH_TSO and FLASH_TS1 registers at the HSI 16 MHz frequency
	18	0x1FFF 0F48	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at the HSI 16 MHz frequency
	19	0x1FFF 0F4C	Store the configuration value of the corresponding FLASH_PERTPE register at the HSI 16 MHz frequency
	20	0x1FFF 0F50	Store the configuration value of the corresponding FLASH_ SMERTPE register at the HSI 16 MHz frequency
	21	0x1FFF 0F54	Store the configuration values of the corresponding FLASH_P RGTPE and FLASH_PRETPE registers at the HSI 16 MHz frequency
	22	0x1FFF 0F58	Store the configuration values of the corresponding FLASH_TSO and FLASH_TS1 registers at the HSI 22.12 MHz frequency
	23	0x1FFF 0F5C	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at the HSI 22.12 MHz frequency
	24	0x1FFF 0F60	Store the configuration value of the corresponding FLASH_PERTPE register at the HSI 22.12 MHz frequency
	25	0x1FFF 0F64	Store the configuration value of the corresponding FLASH_ SMERTPE register at the HSI 22.12 MHz frequency
	26	0x1FFF 0F68	Store the configuration values of the corresponding FLASH_P RGTPE and FLASH_PRETPE registers at the HSI 22.12 MHz frequency
	27	0x1FFF 0F6C	Store the configuration values of the corresponding FLASH_TSO and FLASH_TS1 registers at the HSI 24 MHz frequency
	28	0x1FFF 0F70	Store the configuration values of the corresponding FLASH_TS2P and FLASH_TPS3 registers at the HSI 2 4MHz frequency
	29	0x1FFF 0F74	Store the configuration value of the corresponding FLASH_PERTPE register at the HSI 24 MHz frequency
	30	0x1FFF 0F78	Store the configuration value of the corresponding FLASH_ SMERTPE register at the frequency of HSI 24 MHz frequency
	31	0x1FFF 0F7C	Store the configuration values of the corresponding FLASH_PRGTPE and FLASH_PRETPE registers at the HSI 24 MHz frequency
1	0	0x1FFF 0F80 - 0x1FFF 0FFF	RESERVED

4.4.1. HSI_TRIMMING_FOR_USER

Address offset: 0x1FFF 0F00 to 0x1FFF 0F10

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	HSI_FS[2:0]		
													R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	HSI_TRIM[12:0]												
			R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to read data from this address, and then write to HSI_FS[2:0] and HSI_TRIM[12:0] corresponding to the RCC_ICSCR register to change the HSI frequency.

4.4.2. Calibration value of temperature sensor

Address offset: 0x1FFF 0F14 (30℃), 0x1FFF 0F18 (85℃)

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											
				TS_{CAL}[11:0]											

Software needs to read data from this address.

4.4.3. HSI_4M/8M/16M/22.12M/24M_EPPARA0

Address offset: 0x1FFF 0F1C (4 MHz), 0x1FFF 0F30 (8 MHz), 0x1FFF 0F44 (16 MHz), 0x1FFF 0F58 (22.12 MHz), 0x1FFF 0F6C (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	TS1[8:0]								
							R	R	R	R	R	R	R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TS3[7:0]								TS0[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to set the HSI clock frequency according to the need, choose to read the data from the corresponding address, and then write the FLASH_TS0, FLASH_TS1, FLASH_TS3 registers to realize the configuration of the erasing and programming time required by the corresponding HSI frequency.

4.4.4. HSI_4M/8M/16M/22.12M/24M_EPPARA1

Address offset: 0x1FFF 0F20 (4 MHz), 0x1FFF 0F34 (8 MHz), 0x1FFF 0F48 (16 MHz), 0x1FFF 0F5C (22.12 MHz), 0x1FFF 0F70 (24 MHz)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	TPS3[10:0]										
					R	R	R	R	R	R	R		R	R	R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	Res	TS2P[7:0]							
								R	R	R	R	R	R	R	R

The software needs to set the HSI clock frequency according to the need, choose to read the data from the corresponding address, and then write the FLASH_TS2P and FLASH_TPS3 registers to realize the configuration of the erasing and programming time required for the corresponding HSI frequency.

4.4.5. HSI_4M/8M/16M/22.12M/24M_EPPARA2

Address offset: 0x1FFF 0F24 (4 MHz), 0x1FFF 0F38 (8 MHz), 0x1FFF 0F4C (16 MHz), 0x1FFF 0F60 (22.12 MHz), 0x1FFF 0F74 (24 MHz)

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	PERTPE[16]
															R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PERTPE[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to set the HSI clock frequency according to the need, choose to read the data from the corresponding address, and then write it into the FLASH_PERTPE register to realize the configuration of the erasing and programming time required for the corresponding HSI frequency.

4.4.6. HSI_4M/8M/16M/22.12M/24M_EPPARA3

Address offset: 0x1FFF 0F28 (4 MHz), 0x1FFF 0F3C (8 MHz), 0x1FFF 0F50 (16 MHz), 0x1FFF 0F64 (22.12 MHz), 0x1FFF 0F78 (24 MHz)

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	SMERTPE[16]
															R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SMERTPE[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

The software needs to set the HSI clock frequency according to the need, choose to read the data from the corresponding address, and then write it into the FLASH_SMERTPE register to realize the configuration of the erasing and programming time required for the corresponding HSI frequency.

4.4.7. HSI_4M/8M/16M/22.12M/24M_EPPARA4

Address offset: 0x1FFF 0F2C (4 MHz), 0x1FFF 0F40 (8 MHz), 0x1FFF 0F54 (16 MHz), 0x1FFF 0F68 (22.12 MHz), 0x1FFF 0F7C (24 MHz)

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

The software needs to set the HSI clock frequency according to the need, choose to read the data from the corresponding address, and then write it into the FLASH_PRGTPE and FLASH_PRETPE registers to realize the configuration of the erasing and programming time required for the corresponding HSI frequency.

4.5. Flash protection

The protection of Flash main memory includes the following mechanisms:

- Software design kit (SDK) is used to protect access to specific program areas, and the granularity is 2 kbytes.
- Read protection (RDP) is used to prevent access from outside.
- Wrtie protection (WRP) control is used to prevent unwanted writes (due to confusion of the program memory pointer PC). The granularity of write protection is designed to be 4 kbytes.
- Option byte write protection, special unlocking design.

4.5.1. Flash software development kit (SDK) area protection

The area protected by the FLASH_SDKR register follows the permission description of table 15.

The protection area is defined by SDKR _STRT[4:0], SDKR_END[4:0] of the FLASH_SDKR register, and each bit corresponds to 2 kbytes.

Start address

FLASH memory base address + SDK _STRT[4:0] x 0x800 (included)

End address

FLASH memory base address + (SDK _END[4:0] +1) x 0x800 (excluded)

When SDK _STRT[4:0] is greater than SDK_END[4:0], SDK protection is invalid. When SDK _STRT[4:0] is less than or equal to SDK _END[4:0], SDK protection is effective.

When the protection is in effect, when the FLASH_SDKR register is unprotected (writing SDK _STRT[4:0] is greater than SDK_END[4:0]), the hardware will first trigger mass erase (the protected program in the SDK area has been written before, and the mass erase is used to protect the program in the SDK area), and then the value

of the SDK option in the Flash option byte is updated (the updated value at this time is that the SDK protection is invalid).

At this time, the content of the FLASH_SDKR register will not be updated, until the power-on reset (POR/BOR/PDR) or OBL reset, the register content will be loaded from the SDK option in the Flash option byte into the register.

4.5.2. Flash read protection

By setting RDP option byte, and perform system reset (POR/BOR or OPL reset) to load a new RDP option byte to activate the read protection function. RDP protects main Flah memory, option byte, and SRAM.

If the read protection is set while the debug by SWD is still connected, a power-on reset is required instead of a system reset.

When the RDP option byte and the two's complement code exist in the option byte, the Flash memory will be protected.

Table 4-6 Flash read protection status

RDP byte value	RDP complemented byte value	Read protection level
0xAA	0x55	Level 0
Any value except the combination of (0xAA and 0x55)		Level 1

Regardless of any protection level, system memory is access only and program and erase operations cannot be performed.

Level 0: No protection

To read, program and erase the main Flash memory, as well as any operation to the option byte.

Level 1: Read protection

When the RDP and its two's complement in the option byte contain any combination rather than 0xAA, 0x55, the level 1 read protection takes effect, and the level 1 is the default protection level.

- User mode: The program executed in user mode (boot from main Flash memory) can perform all operations on main Flash and option byte.
- Debug, boot from SRAM and boot from system memory mode (Boot loader): In debug mode, or when booting from SRAM or system memory (Boot loader), the main Flash memory cannot be accessed. In these modes, a bus error is generated for a read or program access to the main Flash, and a hard fault interrupt is generated.

When it is already at Level 1 (any number rather than 0xAA), changing to Level 0 by programming 0xAA, the hardware will perform a mass erase operation on the main Flash memory.

Table 4-7 The relationship between access status and protection level and execution mode

Area	READ Protection level	SDK Area Protection level	Boot From Main Flash(CPU)						Debug/ excuted From RAM/ excuted From System memory			DMA		
			User execution (From Non SDK Area)			User execution (From SDK Area)								
			Read	Write	Erase	Read	Write	Erase	Read	Write	Erase	Read	Write	Erase
Non SDK Area	0	Disable	Yes	Yes	Yes	N/A	N/A	N/A	Yes	Yes	Yes	Yes	No	No
		Enable	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
SDK Area		Disable	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

		En-able	No	No	No	Yes	Yes	Yes	No	No	No	No	No	No
Non SDK Area	1	Dis-able	Yes	Yes	Yes	N/A	N/A	N/A	No	No	No	No	No	No
		En-able	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No
SDK Area		Dis-able	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		En-able	No	No	No	Yes	Yes	Yes	No	No	No	No	No	No
Sys-tem memor y	x	Dis-able	Yes	No	No	N/A	N/A	N/A	Yes	No	No	No	No	No
		En-able	Yes	No	No	Yes	No	No	Yes	No	No	No	No	No
Option bytes area	x	Dis-able	Yes	Yes	Yes	N/A	N/A	N/A	Yes	Yes	Yes	No	No	No
		En-able	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No
Fac-tory bytes	x	Dis-able	Yes	No	No	N/A	N/A	N/A	Yes	No	No	No	No	No
		En-able	Yes	No	No	Yes	No	No	Yes	No	No	No	No	No
UID	x	Dis-able	Yes	No	No	N/A	N/A	N/A	Yes	No	No	No	No	No
		En-able	Yes	No	No	Yes	No	No	Yes	No	No	No	No	No

Note:

(1) Mass erase command issued from any area will erase the SDK area.

(2) Any modification of level 1 to level 0 will trigger the hardware mass erase of the main Flash memory.

(3) The meaning of N/A is that when the SDK Area is disabled, since there is no SDK Area, no situation in which programs can be read out from the SDK Area in the above table, and no situation in which the programs read out from other areas can access the SDK Area.

(4) There are two cases for executing programs from SRAM or system memory: one is Boot from, the other is boot from other memory, and the program jumps to SRAM or system memory.

4.5.3. Flash write protection

Flash can be set to be write-protected against unwanted writes. Define the control granularity of each bit of the WRP register as a write protection (WRP) area of 4 kbytes, that is, the size of 1 sector. See the description of the WRP register for details.

When the WRP area is activated, erase or program operations are not allowed. Accordingly, the mass erase function does not work even if only one area is set as write-protected.

In addition, if an attempt is made to erase or program a write-protected area, the write -protection error flag (WRPERR) of the FLASH_SR register will be set.

Note: Write protection only works on main Flash, and read doesn't work on system memory.

4.5.4. Option byte write protection

By default, Option bytes are readable and write-protected. To gain erase or program access to option bytes, the correct sequence needs to be written to the OPTKEYR register.

4.6. Flash interrupt

Table 4-8 Flash interrupt request

Interrupt event	Event flag	Time stamp/interrupt clear method	Control bit enable
End of operation	EOP	Write EOP = 1	EOPIE
Write protection	WRPERR	Write WRPERR = 1	ERRIE

Note: The following events do not have a separate interrupt flag, but will generate a Hard fault:

- Sequence error of FLASH_CR register of unlock Flash memory.
- Unlock Flash option bytes write sequence error.
- Flash program operation is not aligned with 32-bit data.
- Flash erase (including page erase, sector erase and mass erase) operations do not perform 32-bit data alignment.
- To the option byte register is not aligned with 32-bit data.

4.7. Flash register description

4.7.1. 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
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	Res	Res	Res	Res	Res	Res	Res	Res	LA-TENCY
															RW

Bit	Name	R/W	Reset Value	Function
31:1	Reserved			
0	LATENCY	RW	0	The wait state corresponding to the read operation: 0: There is no wait state for Flash read operation (system clock is 24 MHz and below). 1: The Flash read operation has one wait state, which is two system clock cycles are required for each Flash read (the system clock is at 48 MHz).

4.7.2. Flash key register (FLASH_KEYR)

Address offset: 0x08

Reset value: 0x0000 0000

All register bits are write-only and read as 0.

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

Bit	Name	R/W	Reset Value	Function
31:0	KEY[31:0]	W	0x0000	The following values must be written consecutively to unlock the FLASH_CR register and enable the program/erase operation of the Flash KEY1: 0x4567 0123 KEY2: 0xCDEF 89AB

4.7.3. Flash option key register (FLASH_OPTKEYR)

Address offset: 0x0C

Reset value: 0x0000 0000

All register bits are write-only and read as 0.

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

Bit	Name	R/W	Reset Value	Function
31:0	OPTKEY[31:0]	W	0x0000 0000	The following values must be written consecutively to unlock the option register of the Flash and enable the program/erase operation of the option byte KEY1: 0x0819 2A3B KEY2: 0x4C5D 6E7F

4.7.4. Flash status register (FLASH_SR)

Address offset: 0x10

Reset value: 0x0000 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	Res	Res	BSY
															R
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPTV ERR	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WRP ERR	Res	Res	Res	EOP
RC_W1											RC_W1				RC_W1

Bit	Name	R/W	Reset Value	Function
31:17	Reserved			
16	BSY	R	0	Busy bit This bit indicates that the operation of the Flash is in progress. This bit is set by hardware at the beginning of a Flash operation, and is cleared by hardware when the operation is completed or an error occurs.
15	OPTVERR	RC_W1	0	Option and trimming bits loading validity error when the option and trimming bits and their one's complements do not match. Load unmatched option bytes, coerced to safe values. Software writes 1 to clear.
14:5	Reserved			
4	WRPERR	RC_W1	0	Write protection error This bit is set by hardware when the address to be programmed/erased is in a write-protected Flash region (WRP). Write 1 to clear this bit.
3:1	Reserved			
0	EOP	RC_W1	0	When the program/erase operation of the Flash completes successfully. This bit is only set if the EOPIE bit in the FLASH_CR register is enabled. Write 1 to clear this bit.

4.7.5. Flash control register (FLASH_CR)

Address offset: 0x14

Reset value: 0xC000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	OPTLOCK	Res	Res	OBL_LAUNCH	Res	ERRIE	EOPIE	Res	Res	Res	Res	PGSTRT	Res	OPTSTRT	Res

RS	RS			RC_W1		RW	RW					RW		RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	SER	Res	Res	Res	Res	Res	Res	Res	Res	MER	PER	PG
				RW									RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31	Lock	RS		FLASH_CR Lock bit. Software can only set this bit. When set, the FLASH_CR register is locked. When the unlock timing is successfully given, this bit is cleared by hardware, and the FLASH_CR register is unlocked. The software should set this bit after the program/erase operation is completed. When an unsuccessful unlock sequence is given, this bit remains set until the next system reset.
30	OPTLOCK	RS		Option bytes Lock bit. Software can only set this bit. When set, the bits related to option bytes in the FLASH_CR register are locked. When the unlock timing is successfully given, this bit is cleared by hardware, and the FLASH_CR register is unlocked. The software should set this bit after the program/erase operation is completed. When an unsuccessful unlock sequence is given, this bit remains set until the next system reset.
29:28	Reserved			
27	OBL_LAUNCH	RC_W1		Force the option bytes loading. When set, this bit forces the system to perform a reload of option bytes. This bit is only cleared by hardware when the option byte load has been completed. This bit cannot be written if the OPTLOCK bit is set. 0: Option byte loading completed 1: Option byte loading request is generated, the system resets, and the option byte is reloaded.
25	ERRIE	RW		Error interrupt enable bit, when the WRPERR bit in the FLASH_SR register is set, if this bit is enabled, an interrupt request is generated. 0: No interrupt is generated 1: An interrupt is generated
24	EOPIE	RW		End of operation interrupt enable This bit enables interrupt generation when the EOP bit in the FLASH_SR register is set. 0: EOP interrupt disabled 1: EOP interrupt enable
23:18	Reserved	RW		
19	PGSTRT	RW		The start bit of the program operation of the Flash main memory. Program operation of the main Flash memory, and is set by software. After the BSY bit of the FLASH_SR register is cleared, the hardware clears this bit.
18	Reserved			
17	OPTSTRT	RW		Flash option bytes modified start bit This bit initiates modification of option bytes. Set by software and cleared by hardware after the BSY bit in the FLASH_SR register is cleared. Note: When modifying the Flash option bytes, the hardware will automatically perform the erase operation on the entire page of 128 bytes, and then perform the program operation, which also includes the automatic writing of the two's complement code.
16:12	Reserved			
11	SER	RW		4 kbyte Sector erase operation 0: Sector erase operation of Flash is not selected 1: Select the sector erases operation of Flash Note:

				1） Sector erase will not work on Flash information memory. 2） Sector erase has no effect on areas set to WRP.
10:3	Reserved			
2	MER	RW		Mass erase operation 0: Mass erase operation of Flash is not selected 1: Select the mass erases operation of Flash Note: Mass erase will not work on Flash information memory. Mass erase does not work when WRP is set
1	PER	RW		Page erase operation 0: Page erase operation of the Flash is not selected 1: Select the page erase operation of Flash
0	PG	RW		Program operation 0: Program operation of Flash is not selected 1: Select the program operation of Flash

4.7.6. Flash option register (FLASH_OPTR)

Address offset: 0x20

Reset value: 0x 0000 xxxx

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash in formation memory and written to the corresponding option bit of the register.

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**
nBOOT1	NRST_MODE	WWDG_SW	IWDG_SW	BOR_LEV[2:0]			BOR_EN	RDP[7:0]							
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15	nBOOT1	RW		Select the boot mode with the BOOT PIN
14	NRST_MODE	RW		0: Reset input only 1: GPIO function
13	WWDG_SW	RW		0: Hardware watchdog 1: Software watchdog
12	IWDG_SW	RW		0: Hardware watchdog 1: Software watchdog
11:9	BOR_LEV[2:0]	RW		000: BOR rising threshold is 1.8 V, falling threshold is 1.7 V 001: BOR rising threshold is 2.0 V, falling threshold is 1.9 V 010: BOR rising threshold is 2.2 V, falling threshold is 2.1 V 011: BOR rising threshold is 2.4 V, falling threshold is 2.3 V 100: BOR rising threshold is 2.6 V, falling threshold is 2.5 V 101: BOR rising threshold is 2.8 V, falling threshold is 2.7 V 110: BOR rising threshold is 3.0 V, falling threshold is 2.9 V 111: BOR rising threshold is 3.2 V, falling threshold is 3.1 V
8	BOR_EN	RW		BOR enable 0: BOR is not enabled 1: BOR is enabled, BOR_LEV works
7:0	RDP	RW		0xAA: level 0, read protection inactive Non 0xAA: level 1, read protection active

4.7.7. Flash SDK address register (FLASH_SDKR)

Address offset: 0x24

Reset value: 32'b0000 0000 0000 0000 000X _ XXXX 000X XXXX

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash information memory and written to the corresponding option bit of the register.

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	SA_END[4:0]					Res	Res	Res	SA_STRT[4:0]				
			RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved			
12:8	SDK_END [4:0]	RW		SDK area end addres, each corresponding STEP is 2 kbytes
7:5	Reserved			
4:0	SDK_STRT[4:0]	RW		SDK area start address, each corresponding STEP is 2 kbytes

4.7.8. Flash WRP address register (FLASH_WRPR)

Address offset: 0x2C

Reset value: 0x0000 XXXX

After the power-on reset (POR/BOR/OBL_LAUNCH) is released, the corresponding value is read from the option bytes area of the Flash in formation memory and written to the corresponding option bit of the register.

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**
WRP[15: 0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15	WRP	RW	1	0: Sector 15, with write protection, program and erase are not allowed 1: Sector 15, no write protection
14	WRP	RW	1	0: Sector 14, with write protection, program and erase are not allowed 1: Sector 14, no write protection
13	WRP	RW	1	0: Sector 13, with write protection, program and erase are not allowed 1: Sector 13, no write protection
12	WRP	RW	1	0: Sector 12, with write protection, program and erase are not allowed 1: Sector 12, no write protection
11	WRP	RW	1	0: Sector 11, with write protection, program and erase are not allowed 1: Sector 11, no write protection
10	WRP	RW	1	0: Sector 10, with write protection, program and erase are not allowed 1: Sector 10, no write protection
9	WRP	RW	1	0: Sector 9, with write protection, program and erase are not allowed 1: Sector 9, no write protection
8	WRP	RW	1	0: Sector 8, with write protection, program and erase are not allowed 1: Sector 8, no write protection
7	WRP	RW	1	0: Sector 7, with write protection, program and erase are not allowed 1: Sector 7, no write protection
6	WRP	RW	1	0: Sector 6, with write protection, program and erase are not allowed 1: Sector 6, no write protection
5	WRP	RW	1	0: Sector 5, with write protection, program and erase are not allowed

				1: Sector 5, no write protection
4	WRP	RW	1	0: Sector 4, with write protection, program and erase are not allowed 1: Sector 4, no write protection
3	WRP	RW	1	0: Sector 3, with write protection, program and erase are not allowed 1: Sector 3, no write protection
2	WRP	RW	1	0: Sector 2, with write protection, program and erase are not allowed 1: Sector 2, no write protection
1	WRP	RW	1	0: Sector 1, with write protection, program and erase are not allowed 1: Sector 1, no write protection
0	WRP	RW	1	0: Sector 0, with write protection, program and erase are not allowed 1: Sector 0, no write protection

4.7.9. Flash sleep time configuration register (FLASH_STCR)

Address offset: 0x90

Reset value: 0x0000 6400

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**
SLEEP_TIME[7:0]								Res	Res	Res	Res	Res	Res	Res	SLEEP_EN
RW	RW	RW	RW	RW	RW	RW	RW								RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
15:8	SLEEP_TIME	RW	0x64	FLASH sleep time count (counter based on HSI_10M clock) When the system clock selects LSI or LSE, in order to obtain more optimized power consumption in Run mode, which can use the function of this register (it is only recommended to use this function when LSI or LSE is the system clock). When this function is enabled, the time width of the Flash in the Sleep state in each half system clock low period is: t_{HSI_10M}*SLEEP_TIME Note : t_{HSI_10M} is the period of HSI_10M. To ensure the correct Flash function, the maximum setting value of this register is recommended to be set to 0x28.
7:1	Reserved			
0	SLEEP_EN	RW	0	FLASH Sleep enable 1: Enable Flash sleep 0: Disable Flash sleep

4.7.10. Flash TS0 register (FLASH_TS0)

Address offset: 0x100

Reset value: 0x0000 00B4

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	Res	TS0							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
7:0	TS0	RW	0xB4	HSI with different output frequency, to set the following corresponding values

				HSI is 4 MHz: 0x1E HSI is 8 MHz: 0x3C HSI is 16 MHz: 0x78 HSI is 22.12 MHz: 0xA6 HSI is 24 MHz: 0xB4

4.7.11. Flash TS1 register (FLASH_TS1)

Address offset: 0x104

Reset value: 0x0000 01B0

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	TS1								
							RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:9	Reserved			
8:0	TS1	RW	0x1B0	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x48 HSI is 8 MHz: 0x90 HSI is 16 MHz: 0x120 HSI is 22.12 MHz: 0x18F HSI is 24 MHz: 0x1B0

4.7.12. Flash TS2P register (FLASH_TS2P)

Address offset: 0x108

Reset value: 0x0000 00B4

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	Res	TS2P							
								RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
7:0	TS2P	RW	0xB4	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x1E HSI is 8 MHz: 0x3C HSI is 16 MHz: 0x78 HSI is 22.12 MHz: 0xA6 HSI is 24 MHz: 0xB4

4.7.13. Flash TPS3 register (FLASH_TPS3)

Address offset: 0x10C

Reset value: 0x0000 06C0

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	TPS3										
					RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:11	Reserved			
10:0	TPS3	RW	0x6C0	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x120 HSI is 8 MHz: 0x240 HSI is 16 MHz: 0x480 HSI is 22.12 MHz: 0x639 HSI is 24 MHz: 0x6C0

4.7.14. Flash TS3 register (FLASH_TS3)

Address offset: 0x110

Reset value: 0x0000 00B4

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	Res	TS3							
															RW

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
7:0	TS3	RW	0xB4	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x1E HSI is 8 MHz: 0x3C HSI is 16 MHz: 0x78 HSI is 22.12 MHz: 0xA6 HSI is 24 MHz: 0xB4

4.7.15. Flash page erase TPE register (FLASH_PERTPE)

Address offset: 0x114

Reset value: 0x0001 1940

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	PERTPE
															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PERTPE															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:17	Reserved			
16:0	PERTPE	RW	0x11940	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x2EE0 HSI is 8 MHz: 0x5DC0 HSI is 16 MHz: 0xBB80 HSI is 22.12 MHz: 0x10338 HSI is 24 MHz: 0x11940

4.7.16. Flash sector/mass erase TPE register (FLASH_SMERTPE)

Address offset: 0x118

Reset value: 0x0001 1940

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	SMERTPE
															RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

SMERTPE															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:17	Reserved			
16:0	SMERTPE	RW	0x11940	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x2EE0 HSI is 8 MHz: 0x5DC0 HSI is 1 6 MHz: 0xBB80 HSI is 22.12 MHz: 0x10338 HSI is 24 MHz: 0x11940

4.7.17. Flash program TPE register (FLASH_PRGTPE)

Address offset: 0x11C

Reset value: 0x0000 5DC0

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**
PRGTPE															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15:0	PRGTPE	RW	0x5DC0	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0xFA0 HSI is 8 MHz: 0x1F40 HSI is 16 MHz: 0x3E80 HSI is 22.12 MHz: 0x5668 HSI is 24 MHz: 0x5DC0

4.7.18. Flash pre-program TPE register (FLASH_PRETPE)

Address offset: 0x120

Reset value: 0x0000 12C0

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	PRETPE[13:0]													
		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:14	Reserved			
13:0	PRETPE	RW	0x12C0	HSI with different output frequency, to set the following corresponding values HSI is 4 MHz: 0x320 HSI is 8 MHz: 0x640 HSI is 16 MHz: 0xC80 HSI is 22.12 MHz: 0x1148 HSI is 24 MHz: 0x12C0

4.7.19. Flash register mapping

Offs	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

et																																	
0x00	FLASH_ACR	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.	LA-
	Reset value																																0
0x08	FLASH_KEYR	KEY[31:16]																KEY[15:0]															
	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
0x0C	FLASH_OPTKEYR	OPTKEY[31:16]																OPTKEY[15:0]															
	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
0x10	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BSY	OPTVER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPER	Res.	Res.	Res.	EOP
	Reset value																0	0											0				0
0x14	FLASH_CR	LOCK	OPTLOCK	Res.	Res.	OBL_LAUNC	Res.	ERRIE	EOPIE	Res.	Res.	Res.	Res.	PGSTRT	Res.	OPTSTRT	Res.	Res.	Res.	Res.	Res.	SER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MER	PER	PG
	Reset value	0	0			0								0		0						0									0	0	0
0x20	FLASH_OPTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	nBOOT1	NRST_MODE	WWDG_SW	IWDG_SW	BOR_LEV[2:0]			BOR_EN	RDP[7:0]							
	Reset value																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x24	FLASH_SDKR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SA_END[4:0]					Res.	Res.	Res.	SA_STRT[4:0]				
	Reset value																				X	X	X	X	X				X	X	X	X	X
0x2C	FLASH_WRPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP[15:0]															
	Reset value																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x90	FLASH_STCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEP_TIME[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEP_E
	Reset value																	0	1	1	0	0	1	0	0								0
0x100	FLASH_TS0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS0[7:0]							
	Reset value																									1	0	1	1	0	1	0	0
0x104	FLASH_TS1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS1[8:0]								
	Reset value																								1	1	0	1	1	0	0	0	0

0x108	FLASH_TS2P	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.	TS2P[7:0]											
	Reset value																										1	0	1	1	0	1	0	0				
0x10C	FLASH_TPS3	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.	FLASH_TPS3[10:0]											
	Reset value																										1	1	0	1	1	0	0	0	0	0	0	0
0x110	FLASH_TS3	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.	TS3[7:0]											
	Reset value																										1	0	1	1	0	1	0	0				
0x114	FLASH_PERTPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERTPE	WRP[15:0]																			
	Reset value																	0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0				
0x118	FLASH_SMERTPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMERTPE[16:0]																				
	Reset value																	0	1	1	1	1	1	1	0	1	0	0	1	0	0	0	0	0				
0x11C	FLASH_PRGTPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRGTPE[15:0]																			
	Reset value																		1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0				
0x120	FLASH_PRETPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRETPE[13:0]																	
	Reset value																				0	1	0	0	1	0	1	1	0	0	0	0	0	0				

5. Power control

5.1. Power supply

5.1.1. Power block diagram

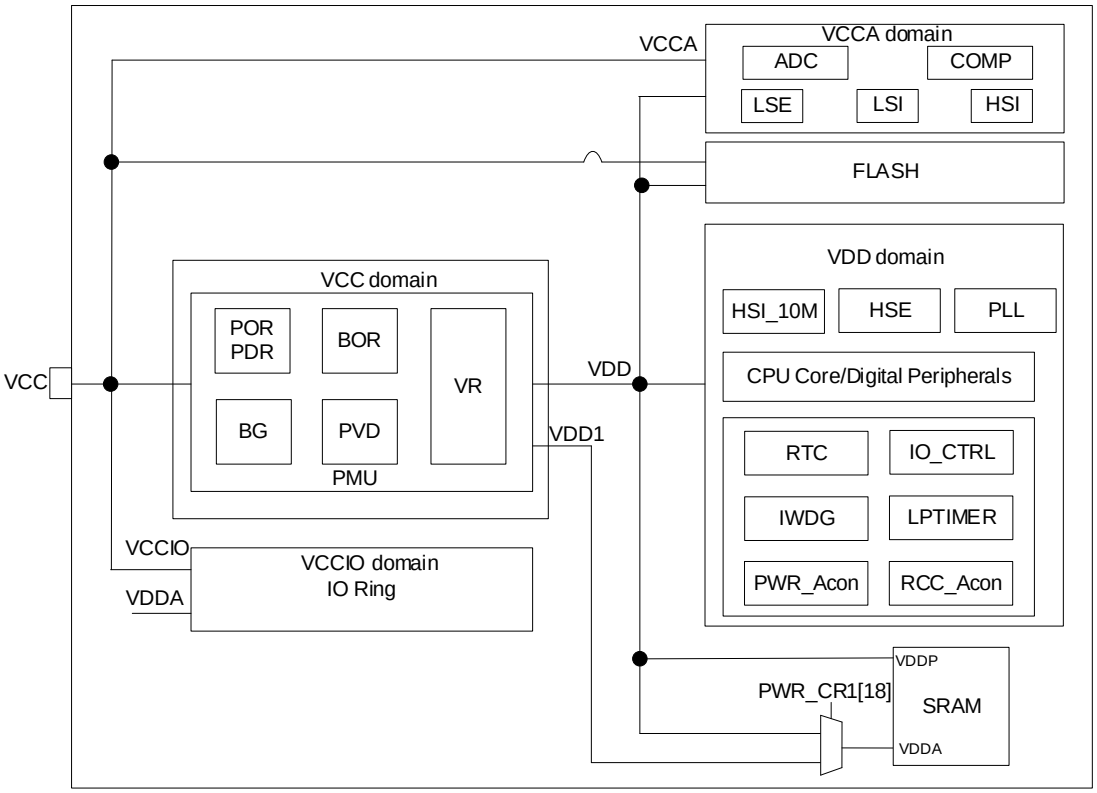


Figure 5-1 Power block diagram

Table 5-1 Power block

Num-bering	Power supply	Power value	Describe
1	VCC	1.7 to 5.5 V	Provide power to the chip through the power pins, and its power supply analog circuit module.
2	VCCA	1.7 to 5.5 V	Provide power to most of the analog modules from VCC PAD (a separate power supply PAD can also be designed).
3	VCCIO	1.7 to 5.5 V	Power supply to IO, from VCC PAD
4	VDD	1.2/1.0 V ± 10%	Output from VR which supplies power to the main logic circuits and SRAM inside the chip. When the MR is powered, it outputs 1.2 V. When entering the stop mode, according to the software configuration, it can be powered by MR or LPR, and the LPR output is determined to be 1.2 V or 1.0 V according to the software configuration.

5.2. Voltage regulator

The microcontroller designs two voltage regulators:

- Main regulator (MR) keeps working when the chip is in normal operating state.
- Low power regulator (LPR) provides a lower power consumption option in stop mode.

VDD comes from MR or LPR depending on the working mode.

In run mode, MR keeps working, outputs is 1.2 V, and LPR is turned off.

In stop mode, power can be supplied from MR or LPR as determined by software. Likewise, it is up to the software to decide whether VDD is 1.2 V or 1.0 V in the case of LPR power supply after entering stop.

5.3. Dynamic voltage value management

Dynamic voltage value management refers to adjusting the output VDD voltage of VR, to obtain corresponding performance and power consumption with different voltages according to application requirements.

- **Range 1: High performance range**

The typically output of MR is 1.2V (VDD), and the system clock frequency can run as fast as 48 MHz.

- **Range 2: Low power range**

Only in stop mode, it is allowed to enter the low power range, and the range only works for LPR.

By default, the typical output of LPR is 1.2 V (VDD). When the VOS bit of the register is set and the chip enters the stop mode, the MR switches to the low power supply (if the software selects the stop mode, it is powered by LPR powered), and LPR is switched to a typical value of 1.0V (VDD). At this time, part of the logic circuit (LPTIMER) in working state can run under LSI or LSE.

When the chip exits the stop mode, the chip restores the MR power supply, and the VOS bit is also cleared by hardware. When the next time enter the stop mode, the VOS bit need to be set by software to ensure the LPR power supply in stop mode is 1.0V, which can get lower power consumption.

5.4. Power monitoring

5.4.1. Power-on reset (POR)/power-down reset (PDR)/brown-out reset (BOR)

The POR/PDR module is designed in the chip and placed under the VDD power domain to provide power-on and power-off reset for the chip. The module keeps working in all modes.

In addition to POR/PDR, BOR (brown out reset) is also implemented. BOR can only be enabled and disabled through the option byte.

When the BOR is turned on, the BOR threshold can be selected by the option byte, and both the rising and falling detection points can be configured individually.

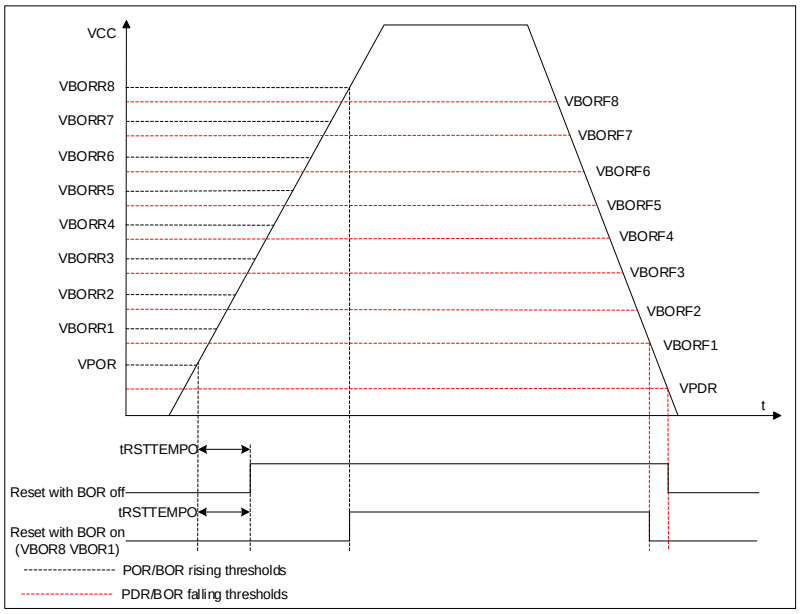


Figure 5-2 POR/PDR/BOR threshold

5.4.2. Programmable voltage detector (PVD)

This module can be used to detect the VCC power supply (also can detect the voltage of the PB7 pin), and the detection point can be configured through the register. When VCC is higher or lower than the detection point of PVD, a corresponding flag is generated.

This event is internally connected to line 16 of EXTI, depending on the rising/falling edge configuration of EXTI line 16, when VCC rises above the detection point of PVD, or VCC falls below the detection point of PVD, an interrupt is generated. In the service program, users can perform urgent shutdown tasks.

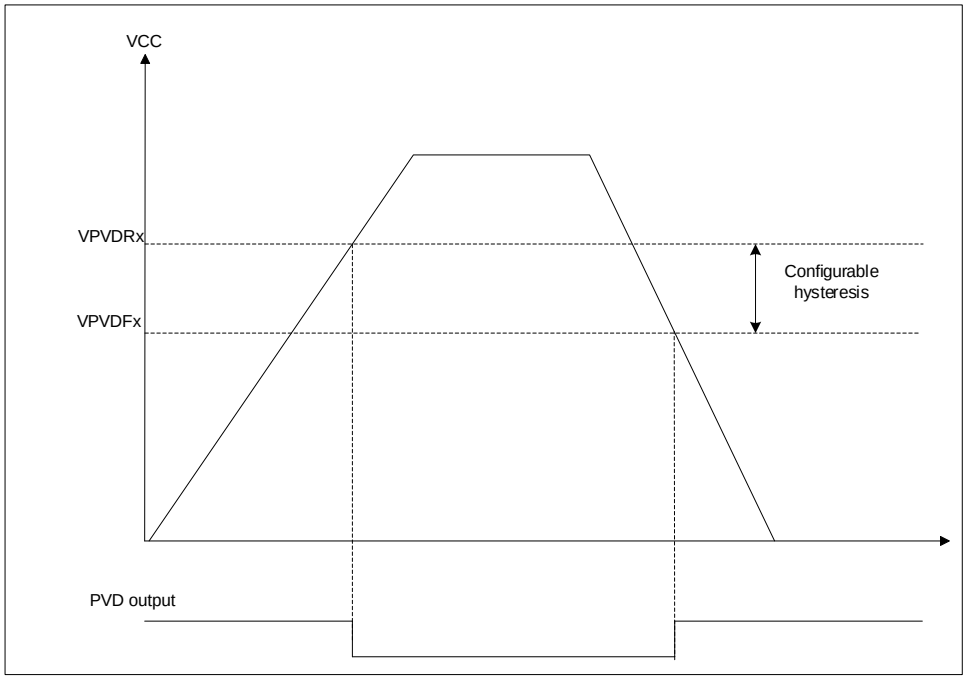


Figure 5-3 PVD threshold

6. Low-power control

By default, the microcontroller is in run mode after a system or a power reset. Several low-power modes are available to save power when the CPU does not need to be kept running, for example when waiting for an external event. Software can choose between power consumption, startup time, and wakeup sources.

6.1. Low-power mode

6.1.1. Introduction to low-power modes

There are two low-power modes:

- **Sleep mode:** The CPU clock is off, NVIC, Sys Tick, peripherals can be configured to keep working. (It is recommended to enable only the modules that must work, and close the modules after the modules work).
- **Stop mode:** In this mode, the contents of SRAM and registers are maintained, the high-speed clock PLL, HSI and HSE are turned off, and the clocks of most modules in the VDD domain are stopped.

In stop mode, LSI, LSE, RTC, LPTIMER, etc. can keep working. For details on the working conditions of each module in this mode, refer to Table 6-2.

In stop mode, the corresponding of VR state can be controlled by software and set to MR or LPR power supply. When LPR is powered, the power consumption greatly reduced, but the wake-up time is long. When the MR power is maintained, the power consumption is large, but it has the ability to wake up quickly in several cycles.

In addition, in run mode, the power consumption can be reduced by the following methods:

- Decrease system clock frequency
- For unused peripherals, turn off peripheral clocks (system clock and module clock)

In summary, the low-power mode transition diagram of this project is as follows.

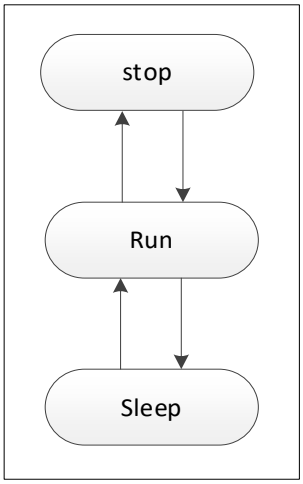


Figure 6-1Low-power mode

6.1.2. Low-power mode switch

Table 6-1 Low power mode switch

Mode	Entry	Wakeup	Wake-up clock	Effects on the clock	Voltage regulator	
					MR	LPR
Sleep (sleep-now or sleep-on-exit)	WFI or Return from ISR	Any interruption	Same as before entering sleep mode	The CPU clock is off and has no effect on other clocks and clock sources.	On(1)	Close
	WFE	Wakeup event				

Stop	SLEEPDEEP bit 1. WFI 2. Return from ISR 3. WFE Note: The system clock cannot select LSI or LSE	Any EXTI line configured to wake up, IWDG, NRST	HSISYS HSI maintains the frequency configuration before entering stop and does not divide the frequency	HSI and PLL are closed. HSE close. LSI and LSE can be turned on or off. LPTIMER, RTC, IWDG: Whether it is configured by software or not. Low-power wakeup and some modules such as RCC keep working. Clocks of the remaining modules are turned off.	Software configuration switch	Software configuration switch, if open, output voltage 1.2/1.0 V can be configured

Note: The software must configure the VR state as MR mode to enter sleep mode.

6.1.3. Functions in each working mode

Table 6-2 Functions in each working mode(1)

Peripheral	Run	Sleep	Stop	
			VR@LPR or VR@MR	Wakeup ability
CPU	Y	-	-	-
Flash memory	Y	Y	-(2)	-
SRAM	Y	O(3)	-(4)	-
Brown-out reset (BOR)	Y	Y	O	O
PVD	O	O	O	O
DMA	O	O	-	-
HSI	O	O	-	-
HSE	O	O	-	-
LSI	O	O	O	-
LSE	O	O	O	-
PLL	O	O	-	-
HSE Clock Security System (CSS)	O	O	-	-
LSE Clock Security System (CSS)	O	O	O	O
RTC	O	O	O	O
USART1	O	O	-	-
USART2	O	O	-	-
I2C	O	O	-	-
SPI1	O	O	-	-
SPI2	O	O	-	-
ADC	O	O	-	-
COMP1/COMP2	O	O	O	O
Temperature sensor	O	O	-	-
Timers(TIM1/TIM3 /TIM14/TIM16/TIM17)	O	O	-	-
LPTIM	O	O	O	O
IWDG	O	O	O	O
WWDG	O	O	-	-
SysTick timer	O	O	-	-
CRC	O	O	-	-
GPIOs	O	O	O	O

Note:

(1) Y = Yes (enable), O = Optional (default disabled, can be enabled by software), - = Not available.

(2) Flash is not powered off, but no clock is provided, and it enters the lowest power consumption state.

(3) SRAM clock can be turned on or off.

(4) SRAM is not powered off, but no clock is provided, and it enters the lowest power consumption state.

(5) Before entering stop mode, if LSE CSS is enabled, when a problem with LSE CSS, it will wake up the system and enter the NMI interrupt.

6.2. Sleep mode

6.2.1. Entering sleep mode

The sleep mode is entered by executing the WFI (wait for interrupt) or WFE (wait for event) instructions. Two option are available to select the sleep mode entry mechanism, depending on the SLEEPONEXIT bit in the Cortex M0+ System Control Register.

- Sleep-now: If the SLEEPONEXIT bit is 0, to enter sleep mode as soon as WFI or WFE instruction is excuted.
- Sleep-on-exit: If the SLEEPONEXIT bit is 1, to enter sleep mode as soon as it exits the low priority ISR.

In the sleep mode, all IO pins keep the same state as in the run mode.

6.2.2. Exiting sleep mode

If the WFI instruction is used to enter sleep mode, any peripheral interrupt acknowledged by NVIC can wake up the device from sleep mode.

If the WFE instruction is used to enter sleep mode, the MCU exits sleep mode as soon as an event occurs. The wakeup events can be generated in the following ways:

- Enable interrupts in the peripheral control register but not in the NVIC, and enabling the SEVONPEND bit in the Cortex M0+. When the MCU resumes from WFE, the peripheral interrupt pending bit and the peripheral NVIC IRQ channel pending bit (in the NVIC interrupt clear pending register) must be cleared.
- Or configuring an external or internal EXTI line in event mode. When the CPU resumes from WFE, it is not necessary to clear the peripheral interrupt pending bit or the NVIC IRQ channel pending bit corresponding to the event line is not set.

This mode offer the shortest wakeup time, and no time is wasted in interrupt entry and exit.

Table 6-3 Sleep-now

Sleep-now mode	Description
Mode entry	WFI or WFE while: - SLEEPDEEP = 0 and - SLEEPONEXIT = 0
Mode exit	Enter the sleep mode through WFI, the exit method is: interrupt. Enter the sleep mode through WFE, the exit method is: wakeup event.
Wakeup latency	None

Table 6-4 Sleep-on-exit

Sleep-on-exit	Description
Mode entry	WFI while: - SLEEPDEEP = 0 and - SLEEPONEXIT = 1
Mode exit	Interrupt
Wakeup latency	None

6.3. Stop mode

The stop mode is based on the Cortex-M0+ deep sleep mode combined with peripheral clock gating, and the VR can be configured as MR or LPR power supply. In stop mode, PLL, HSI and HSE are turned off, SRAM and register contents are kept in a state, LSI, LSE, LPTIMER, RTC, IWDG can be configured by software whether to work, low-power wakeup and some RCC logic and so on, the clock inputs to the digital blocks of the remaining VCORE domains are turned off.

In the stop mode, all IO pins keep the same state as in the run mode.

6.3.1. Entering stop mode

To further reduce power consumption in stop mode, when PWR_CR.LPR = 1, VR can enter LPR to supply power.

If Flash memory programming is ongoing, the stop mode entry is delayed until the memory access is finished (the BSY bit of the FLASH_SR register is read by software to determine whether the current erase and program operations have been completed).

If an access to the APB domain is ongoing, the stop mode entry is delayed until the APB access is finished (controlled by software).

6.3.2. Exiting stop mode

When exiting stop mode by issuing an interrupt or a wakeup event, the HIS oscillator is selected as system clock.

In stop mode, if VR is in LPR state, there is an additional stabilization delay for wakup in stop mode.

In stop mode, if VR is in MR state, the current consumption will be large, but the wakeup time will be reduced.

Table 6-5 Stop mode

Stop mode	Description
Mode entry	WFI (wait for interrupt) or WFE (wait for event) while: Configuration settings: 1）Configure the LPR bit of PWR_CR, to select VR to work under MR or LPR. 2）Configure the VOS bit of PWR_CR, to select LPR mode to provide 1.2 V or 1.0 V. 3）Configure the SRAM_RETV bit of PWR_CR, to select the retention voltage of SRAM. 4）Configure the MRRDY_TIME and FLS_SLPTIME of PWR_CR, to set wake-up time of MR and Flash. Set the SLEEPDEEP bit of Cortex M0+ **Note:** To enter stop mode, all EXTI line pending bits (EXTI_PR register), all peripheral interrupt pending bits and RTC alarm flags must be reset. OtheRWise, the stop mode entry proceduce is ignored and program execution continues. If the application needs to disable HSE before entering stop mode, the system clock source must be first switched to HSI and then clear the HSEON bit. To make the change of chip power consumption as balanced as possible, the software needs to follow the principle of gradual shutdown: gradually shut down the clock of each module, select HSI as the system clock, close PLL and HSE. To shorten the wakeup time, before entering the stop mode, the system clock should be configured to select the HSI high-frequency clock, and the HPRE of the RCC_CFGR register is set to 0, otheRWise the hardware switching clock after wake-up will consume extra clocks.
Mode exit	If using WFI to enter stop mode: - Any EXTI Line configured in interrupt mode (the corresponding EXTI interrupt vector must be enabled in the NVIC). If using WFE to enter stop mode: - Any EXTI Line configured in event mode. - Interrupt pending bit when the CPU SEVONPEND bit is set.
Wakeup latency	LPR to MR wakeup time + HSI wakeup time + Flash wakeup time

6.4. Decreasing system clock frequency

In run mode, the frequency of the system clock (SYSCLK, HCLK, PCLK) can be reduced by frequency division through the prescaler register configuration. These prescalers can also be used to reduce the frequency of peripherals before entering sleep mode.

When the system runs at a lower frequency (32.768 kHz), to obtain less power consumption, software can set the drive capability configuration bit of the voltage regulator (MR) (PWR_CR1 register BIAS_CR [3:0]), which greatly reduces the power consumption of MR itself. But it should be noted that the system clock frequency should be reduced first, and then the driving capability of the MR should be adjusted. On the contrary, when

exiting the lower frequency and entering the higher operating frequency, it need increase the driving capacity of the MR first, and then change the operating frequency of the system clock.

6.5. Peripheral clock gating

In run mode, the AHB clock (HCLK) and APB clock (PCLK) for individual peripherals and memories can be stopped at any time to reduce power consumption.

To reduce the power consumption in sleep mode, peripheral clocks can be stopped before executing WFI or WFE instructions.

6.6. Power management register

The peripheral 's registers can be accessed through half-word or word.

6.6.1. Power control register 1 (PWR_CR1)

Address offset: 0x00

Reset value: 0x0003 0000 (reset by POR)

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	HSION_CTRL	SRAM_RETV[2:0]		
												RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	LPR	FLS_SLPTIME[1:0]		MRRDY_TIME[1:0]		VOS	DBP	Res	Res	Res	BIAS_CR_SEL	BIAS_CR[3:0]			
	RW	RW	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:20	Reserved	-	-	Reserved
19	HSION_CTRL	RW	0	HSI turns on time control when wakeup from stop mode. 0: After waiting for MR to stabilize, enable HIS. 1: Turn on the VR, as well as enable HSI when wakeup
18:16	SRAM_RETV _ [2:0]	RW	111	SRAM retention voltage control in stop mode 000: Reserved 001: Reserved 010: Reserved 011: Supply 0.9 V voltage to SRAM 1xx: Supply 1.2 V or 1.0 V to SRAM (depending on VOS bit)
15	Reserved			
14	LPR	RW	0	Low power regulator 0: Main regulator works in stop mode 1: Low power regulator works in stop mode
13:12	FLS_SLPTIME	RW	2'b00	Wakeup sequence from stop mode, after the HSI is stable, a waiting time is required before the Flash operation. 2'b00: 5 us 2'b01: 2 us 2'b10: 3 us 2'b11: 0 us Note: When this register is set to 2'b11, it means that the program is executed from SRAM instead of Flash after wakeup. And the program guarantees that Flash will not be accessed within 3 us after waking up the execution program.

11:10	MRRDY_TIME	RW	2'b00	During the stop mode, the VDD voltage is LP-VR, and the time control of switching from LP-VR to stable Main-VR during wakeup. 2'b00: 2 us 2'b01: 3 us 2'b10: 4 us 2'b11: 5 us
9	VOS	RW	0	Voltage scaling range selection 0: After entering stop mode, VDD = 1.2 V 1: After entering stop mode, VDD = 1.0 V
8	DBP	RW	0	RTC write protection disabled After reset, the RTC is write-protected to prevent accidental writes. To access the RTC this bit must be set to 1. 0: Disable access to RTC 1: RTC can be accessed
7:5	Reserved	-	-	Reserved
4	BIAS_CR_SEL	RW	0	Select the MR bias current from the configuration of the BIAS_CR register or from the loading of the Factory config.bytes area of the information memory 0: Select the load from the Factory config.bytes area 1: Select from BIAS_CR register
3:0	BIAS_CR	RW	4'b0000	MR bias current configuration. 4'b0000: …

6.6.2. Power control register 2 (PWR_CR2)

Address offset: 0x04

Reset value: 0x0000 0500 (reset by POR)

Note: This register is related to PVD function.

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	FLT_TIME[2:0]			FLTEN	Res	PVDT[2:0]			Res	SRCSEL	Res	PVDE
				RW			RW		RW				RW		RW

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11:9	FLT_TIME	RW	3'b010	Digital filter time configuration The filter time is about 30.7 ms (1024 LSI or LSE clocks) 101: The filter time is about 3.8 ms (128 LSI or LSE clocks) 100: The filter time is about 1.92 ms (64 LSI or LSE clocks) 011: The filter time is about 480 us (16 LSI or LSE clocks) 010: The filter time is about 120 us (4 LSI or LSE clocks) 001: The filter time is about 60 us (2 LSI or LSE clocks) 000: The filter time is about 30 us (1 LSI or LSE clock)
8	FLTEN	RW	1	Digital filter function enable control 0: Disable 1: enable
7	Reserved			
6:4	PVDT[2:0]	RW	000	Voltage rising edge detection threshold (falling edge detection threshold correspondingly reduced by 0.1 V) and PVDIN detection control. 000: VPVD0 (around 1.8 V) 001: VPVD1 (around 2. 0 V) 010: VPVD2 (around 2. 2 V) 011: VPVD3 (around 2.4 V) 100: VPVD4 (around 2.6 V) 101: VPVD5 (around 2.8 V) 110: VPVD6 (around 3.0 V) 111: VPVD7 (around 3.2 V)
3	Reserved			
2	SRCSEL	RW	0	PVD detects power supply selection. 0:VCC 1:Detect PB7 pin

				If this bit is set to 1, the voltage on PB7 is internally compared to VREFINT (including rising and falling thresholds). In this case, the setting of the PVDT register is invalid.
1	Reserved	-	-	Reserved
0	PVDE	RW	0	Voltage detect enable bit 0: Voltage detection disable 1: Voltage detection enable If SYSCFG_CFG2.PVD_LOCK = 1, PVDE is write protected. Write protection is reset only after a system reset.

6.6.3. Power status register (PWR_SR)

Address offset: 0x14

Reset value: 0x0000 0000 (reset by POR)

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	PVDO	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
				R											

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11	PVDO	R		PVD test result 0: The detected VCC or PB7 exceeds the PVD selected compare threshold 1: Detected VCC or PB7 is below the PVD selected compare threshold
10:0	Reserved	-	-	Reserved

6.6.4. PWR register map

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	PWR_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSION_CTRL	SRAM_RETV[2:0]			Res.	LPR	FLS_SLP-TIME[1:0]		MRRD_TIME[1:0]		VOS	DBP	Res.	Res.	Res.	BIAS_CR_SEL	BIAS_CR[3:0]			
	Reset value													0	1	1	1		0	0	0	0	0	0	0				0	0	0	0	0
0x004	PWR_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLT_TIME[2:0]			FLTEN	Res.	PVDT[2:0]			Res.	SRCSEL	Res.	PVDE
	Reset value																					0	1	0	1	1	0	0	0		0		0
0x008	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.
0x00C	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.
0x010	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.	
0x	PWR_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDO	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

14	Reset value																		0												

7. Reset

There are two types of resets defined as power reset and system reset.

7.1. Reset source

7.1.1. Power reset

A power reset sets all registers to their reset value, which occurs in the following situations:

- Power on reset (POR/ PDR)
- Brown-out reset (BOR)

7.1.2. System reset

A system reset sets most registers to their reset values, except some special registers, such as the reset flag register.

A system reset generates when the following events occur:

- Reset of NRST pin
- Windowed watchdog reset (WWDG)
- Independent watchdog reset (IWDG)
- SYSRESETREQ software reset
- Option byte load reset (OBL)
- Power reset (POR/PDR, BOR)

The reset source can be identified by checking the reset flag bits of the RCC_CSR register.

7.1.3. NRST pin (external reset)

By loading the option byte (NRST_MODE bit), the NRST pin can be configured in the following modes (see option byte description for specific configuration):

- Reset input

In this mode, any valid reset signal on the NRST pin is passed to the internal logic, but the reset generated inside the chip is not output on the NRST pin.

In this configuration mode, the PF2 function of the GPIO is invalid.

There is burr filtering for NRST pin. The design ensures that NRST must meet the minimum width of 20 us, and the signal less than this width will be filtered out.

- GPIO

In this mode, the PIN can be used as a standard GPIO, like PF2. The reset function on the pin does not work. Resets are only generated internally by the chip and cannot be passed to the pin.

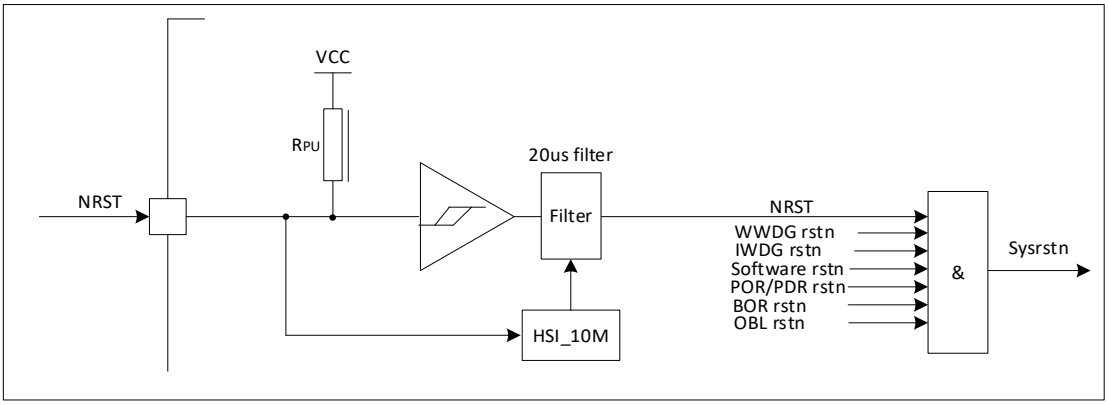


Figure 7-1Simplified diagram of the reset circuit

7.1.4. Watchdog reset

See independent watchdog and system windows watchdog for details.

7.1.5. Software reset

A software reset can be achieved by setting the SYSRESETREQ bit in the ARM M0+ interrupt and reset control register.

7.1.6. Option byte loader reset

By configuring FLASH_CR.OBL_LAUNCH = 1, the software generates an option byte load reset, thereby starting the option byte load again.

8. Clock

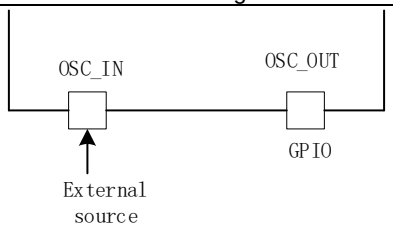
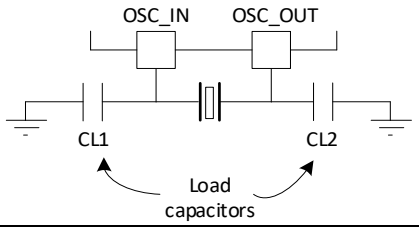
8.1. Clock source

8.1.1. High-speed external clock (HSE)

The high-speed external clock comes from two sources:

- External crystal (crystal), with the internal start-up circuit, a clock signal of 4 to 32 MHz is generated.
- Input high-speed clock source directly from external.

Table 8-1HSE/LSE clock sources

Clock source	Hardware configuration
External clock	
External crystal	

External crystal

The 4 to 32 MHz crystals have very high accuracy. The HSERDY flag of RCC_CR shows whether the HSE is stable. HSE can be turned on or off by the HSEON bit.

External clock source (HSE bypass)

In this mode, the external clock source is directly supplied to the chip. Software selects this mode by the HSEBYP and HSEON bits of RCC_CR. The external clock source will be input into the chip through PF0, and PF1 is used as GPIO.

8.1.2. Low-speed external clock (LSE)

The low-speed clock external (LSE) comes from two sources:

- Through an external crystal (crystal), with the internal start-up circuit, a clock signal of 32.768 kHz is generated.
- Input high-speed clock source directly from external.

SERDY flag in the RCC_BDCR register shows whether the LSE is stable. LSE can be turned on or off by the LSEON bit. The drive capability can be adjusted via LSEDRV[1:0] to obtain a compromise between robustness and short startup time.

External clock source (LSE bypass)

In this mode, an external clock source is provided. This mode is selected by software through the LSEBYP and LSEON bits of RCC_CR. The external clock source is input into the chip through PA10, and PA9 is used as GPIO.

8.1.3. High-speed internal clock (HSI)

The high-speed internal clock is the most important source of the chip system clock. The center frequency of the HSI clock source is designed to be 24 MHz.

8.1.4. Low-speed internal clock (LSI)

The low-speed internal clock, as the clock for RTC, IWDG and LPTIM, and as the system clock when the chip is running at low speed. The clock center frequency is designed at 32.768 kHz.

8.1.5. PLL

The PLL can be used to multiply the frequency of HSI or HSE. Before enabling the PLL, the PLL must be configured. Once the PLL is enabled, these configured registers cannot be changed.

8.2. Clock tree

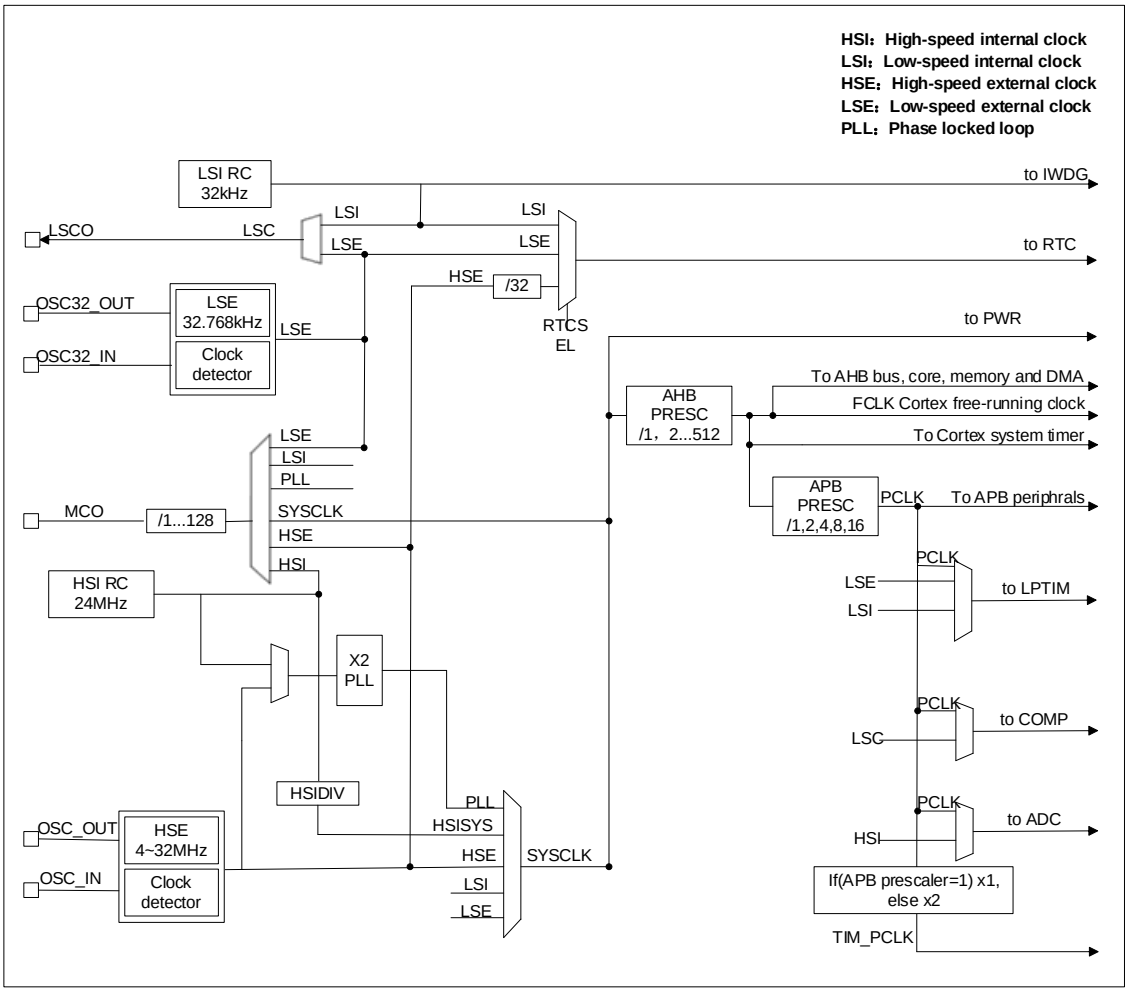


Figure 8-1 System clock structure

8.3. Clock security system (CSS)

The clock security system can be activated by software. In this case, the clock detection is enable after the HSE oscillator startup delay, and disabled when this oscillator is stopped.

If a failure is detected on the HSE clock, the HSE oscillator is automatically disable, a clock failure event is sent to the break input of advanced-control timer (TIM1) and general purpose timers (TIM16/TIM17) and an interrupt

is generated to inform the software of the failure (Clock Security System Interrupt CSSI), which allows the MCU to perform rescue operations. CSSI is linked to the Cortex-M0+ NMI (Non-maskable interrupt) exception vector.
Note: Once the CSS is enabled and if the HSE clock fails, the CSS interrupt occurs and an NMI is automatically generated. The NMI will be executed indefinitely unless the CSS interrupt pending bit is cleared. Therefore, in the NMI ISR user must clear the CSS interrupt by setting the CSSC bit in the clock interrupt register (RCC_CICR).
If the HSE oscillator is used directly or indirectly as the system clock (indirectly means: it is used as PLL input clock, and the PLL clock is used as system clock), a detected failure cause a switch of the system clock to the HIS oscillator and the disabling of the HSE oscillator. If the clock fails, HSE is the input clock to the PLL, the PLL will also disabled.

8.4. Clock-out capability

In order to facilitate board-level applications, save BOM costs and debug requirements, the chip needs to provide a clock output function. That is, the MCO signal (parallel frequency division) in the following table is used to realize the clock output function through the multiplexing function of GPIO.

Table 8-2 Output clock selection

Clock source	MCO output clock source
HSI	√
SYSCLK	√
HSE	√
PLL	√
LSE	√
LSI	√

Note: When switching the MCO clock source and selecting the GPIO AF function as the initial stage of the MCO, the MCO may generate glitches, and this period of time needs to be avoided.

8.5. Internal and external clock calibration with TIM14

Due to factors of temperature, voltage, process and production, the frequency of internal clock sources (such as HSI, LSI, etc.) drifts. Therefore, it is necessary to take some necessary measures to calibrate the frequency drift according to the change of the external working environment of the system.
The basic idea of clock drift processing is: when the external environment of the system changes, the internal clock of the chip is dynamically measured in real time to detect and find problems. Then, the trimming parameters of the internal clock are fine-tuned by software to achieve the purpose of dynamic calibration.

8.5.1. HSI calibration

HSI clock calibration is divided into two parts: clock detection and clock calibration.

Clock measurement

The rationale is based on relative measurements (such as the ratio of HSI/LSE), and the accuracy is closely related to the ratio of the two clock sources. The higher the ratio, the better the measurement.
HSI clock counts between consecutive edges of the LSE signal. Using the high accuracy (ppm level) of the LSE, the user can measure the clock frequency with the same resolution and can compensate for production, process, temperature and voltage related frequency deviations by fine-tuning the clock source.

HSI oscillators have dedicated calibration bits for this purpose and are user accessible. If LSE is not available, select HSE/32 for the most accurate calibration possible. The frequency of the HSI is measured by capturing the signal through the channel 1 input of the TIM14.

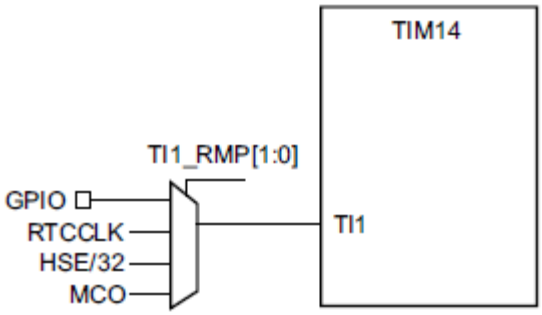


Figure 8-2Frequency measurement with TIM14 in capture mode

The input capture channel of TIM14 can be a GPIO or an in-chip clock. The selection of these clocks is implemented through the TI1_RMP [1:0] register of TIM14_OR. The four options are as follows:

- TIM14 channel 1 is connected to GPIO
- TIM14 channel 1 is connected to RTC Clock
- TIM14 channel 1 is connected to HSE/32 Clock
- TIM14 channel 1 is connected to MCO (Microcontroller clock output)

Clock division

Once the abnormality of HSI clock is detected, it will notify the software to deal with it through interrupt. The software achieves the purpose of dynamic calibration by fine-tuning the trimming parameters of the internal clock.

Connect LSE to the input capture of TIM14 channel 1 through MCO multiplexer, its main purpose is to measure HSI accurately (in this case, HSI should be set as the system clock source). Such a mechanism provides a measure of the internal clock period by counting the number of HSI clocks during two consecutive LSE signal transition edges.

the LSE high precision (ppm) when an external crystal is used, so that it is possible to determine the internal clock frequency with the same resolution, and then trim the clock source to compensate for the frequency drift related to process, temperature and voltage.

HSI is therefore designed with special user-accessible calibration register bits.

The rationale for this implementation mechanism is a relative measure (eg, the ratio of HSI/LSE): the accuracy is thus closely related to the ratio of the frequencies of the two clock sources. The higher the ratio, the better the measure.

8.5.2. LSI calibration

Like HSI, the clock frequency of LSI is also affected by voltage, temperature, process and production drift. The calibration of LSI adopts HSE or HSI whose frequency differs greatly from the calibration, and the calibration method is similar to that of HSI.

The calibration of the LSI is to connect the output of the LSI and the input capture of the TIM14. Define the HSE as the system clock source, and provide a measure of the LSI cycle in the number of HSE clocks of two consecutive LSIs.

In principle, it is still the relationship between relative frequencies, that is, the frequency ratio of HSE/LSI: the calibration accuracy is closely related to this. The larger the ratio value, the better the measurement.

8.6. Reset/clock register

The registers of this module can be accessed with word (32-bit), half-word (16-bit) and byte (8-bit).

8.6.1. Clock control register (RCC_CR)

Address offset: 0x00

Reset value: 0x0000 0100

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	PLL RDY	PLL ON	Res	Res	Res	Res	CSS ON	HSE BYP	HSE RDY	HSE ON
						R	RW					RS	RW		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	HSIDIV[2:0]			HSI RDY	HSI KERON	HSION	Res	Res	Res	Res	Res	Res	Res	Res
		RW			R	RW	RW								

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25	PLLRDY	R	0	PLL clock ready flag. Set by hardware to indicate that the PLL clock is locked 0: PLL unlocked 1: PLL locked
24	PLLON	RW	0	PLL enabled. It can be set and cleared by software. This bit is cleared by hardware when entering stop mode. This bit cannot be re-set if the PLL clock is used as the system clock. 0: PLL OFF 1: PLL ON
23:20	Reserved	-	-	Reserved
19	HSE_CSSON	RS	0	Clock safety system enabled. Set by software to enable the clock security system. When this bit is set, if HSE is ready, the hardware will perform clock detection. When the clock is found to be invalid, the hardware disables the clock detection. This bit can only be set and cleared by reset. 0: Clock security system OFF (clock detection OFF) 1: Clock safety system ON (clock detection ON if HSE is stable, otheRWise OFF)
18	HSEBYP	RW	0	Bypass HSE to connect external crystal, select the pin input clock. It is set and cleared by software, the circumstance of by-passing the external crystal, and connecting the external pin to input the clock. The external clock must be enabled with HSEON. The HSEBYP bit is set only when the HSE external crystal is disabled. 0: HSE external crystal is not dropped by bypass 1: HSE external crystal is bypassed, and external pin input clock
17	HSERDY	R	0	HSE clock ready flag Set by hardware, indicating that the HSE is stable. 0: HSE not ready 1: HSE ready Note: When HSEON is cleared, HSERDY is cleared imme-diately
16	HSEON	RW	0	HSE clock enable Software can be set and cleared. Enter stop mode, the hardware clears this bit. This bit cannot be reset if HSE is used directly or indirectly as the system clock. 0: HSE OFF 1: HSE ON

15:14	Reserved	-	-	Reserved
13:11	HSIDIV[2:0]	RW	0	HSI clock division factor. Software controls these bits to set the frequency division factor of HSI to generate the HSISYS clock 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128
10	HSIRDY	R	0	HSI clock ready flag. Set by hardware to indicate HSI OSC is stable. This bit is only valid when HSION = 1. 0: HSI OSC not ready, 1: HSI OSC ready, When HSION is cleared, HSIRDY will be pulled low immediately.
9	Reserved	-	-	Reserved
8	HSION	RW	1	HSI clock enable bit. Software can set and clear this bit. When entering s top mode, the hardware clears this bit to stop HSI. When the HSI is used directly or indirectly as the system clock (also when exiting stop mode, or when the HSE is used as the system clock and fails). 0: HSI OFF 1: HSI ON
7:0	Reserved	-	-	Reserved

8.6.2. Internal clock source calibration register (RCC_ICSCR)

Address offset: 0x04

Reset value: 0x00FF 10FF, reset by POR/BOR

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res	Res	Res	Res	LSI_STARTUP		Res	LSI_TRIM[8:0]								
				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**
HSI_FS[2:0]			HSI_TRIM[12:0]												
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:28	Reserved		-	Reserved
27:26	LSI_STARTUP	RW	2'b00	Low-speed internal clock (LSI) stabilization time selection: 11: 256 LSI clock cycles 10: 64 LSI clock cycles 01: 16 LSI clock cycles 00: 4 LSI clock cycles
25	Reserved			
24:16	LSI_TRIM	RW	0x0FF	Low-speed internal clock frequency calibration. After power-on, the chip hardware will write the factory information (stored in 0x1FFF 0FA4) into this register, so that the LSI can output an accurate 32.768 kHz frequency. By rewriting the value of this register, the software increases (decrease) the output frequency of LSI by about 0.2% for each increase (decrease) by 1.
15:13	HSI_FS	RW	3'b000	HSI frequency selection: 000: 4 MHz 001: 8 MHz 010: 16 MHz 011: 22.12 MHz 100: 24 MHz > = 101: 4 MHz

				After power-on, 4 MHz is selected by default. After the option byte load is completed, the hardware switches to 8 MHz.
12:0	HSI_TRIM	RW	0x10FF	Clock frequency calibration value. After power-on, the hardware uses the default calibration value of HSI 4 MHz, and the factory information (stored in 0 x1FFF 0FA0) will be written into this register when trimming. The software reads out the data stored in the corresponding address in the information area and writes it into the register to realize the calibration under the specific output frequency of the HSI. Save it in the following address of Flash: 24 MHz calibration value storage Address offset: 0x1FFF 0F10 22.12 MHz calibration value storage Address offset: 0x1FFF 0F0C 16 MHz calibration value storage Address offset: 0x1FFF 0F08 8 MHz calibration value storage Address offset: 0x1FFF 0F04 4 MHz calibration value storage Address offset: 0x1FFF 0F00 After writing the calibration value to this register, the value of this register can also be the central value, and the value of this register can be modified. For each increase (decrease) by 1, the output frequency of the HSI will increase (decrease) by about 0.1%.

8.6.3. Clock configuration register (RCC_CFGR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	MCOPRE[2:0]			Res	MCOSEL[2:0]			Res	Res	Res	Res	Res	Res	Res	Res
	RW				RW										
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	PPRE[2:0]			HPRE[3:0]				Res	Res	SWS[2:0]			SW[2:0]		
	RW			RW						R			RW		

Bit	Name	R/W	Reset Value	Function
31	Reserved	-	-	Reserved
30:28	MCOPRE[2:0]	RW	0	Microcontroller clock output (MCO) frequency division factor. Software controls these bits to set the division factor of the MCO output: 000: 1 001: 2 010: 4 011: 8 100: 16 101: 32 110: 64 111: 128 Set these bits before enabling the MCO output.
27	Reserved	-	-	Reserved
26:24	MCOSEL[2:0]	RW	0	MCO selection 000: No clock, MCO output disabled 001: SYSCLK 010: HSI_10M 011: HIS 100: HSE 101: PLL CLK 110: LSI 111: LSE Note: Incomplete output clock conditions may occur during the clock startup or switchover phase.
23:15	Reserved	-	-	Reserved

14:12	PPRE[2:0]	RW	0	This bit is controlled by software. To generate the PCLK clock, it sets the division factor of HCLK as follows: 0xx: 1 100: 2 101: 4 110: 8 111: 16
11:8	HPRE[3:0]	RW	0	AHB clock division factor. Software controls this bit. In order to generate the HCLK clock, it sets the frequency division factor of SYSCLK as follows: 0xxx: 1 1000: 2 1001: 4 1010: 8 1011: 16 1100: 64 1101: 128 1110: 256 1111: 512 In order to ensure the normal operation of the system, it is necessary to configure an appropriate frequency according to the VR power supply. Note: It is recommended to switch the frequency division factor step by step.
7:6	Reserved	-	-	Reserved
5:3	SWS[2:0]	R	0	System clock switch status bits These bits are controlled by hardware and indicate which clock source is currently being used as the system clock: 000: HSISYS 001: HSE 010: PLL CLK 011: LSI 100: LSE Others: Reserved
2:0	SW[2:0]	RW	0	System clock source selection bits. Controlled by software and hardware, these bits select the system clock: 000: HSISYS 001: HSE 010: PLL CLK 011: LSI 100: LSE Others: Reserved The hardware is configured as HSISYS include: 1) The system exits from stop mode 2) Software configuration 001 (HSE), HSE failure occurs (HSE is the system clock source, or HSE is the PLL input, and PLL is the system clock source)

8.6.4. PLL configuration register (RCC_PLLCFGR)

Address offset: 0x0C

Reset value: 0x0000 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	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	Res	Res	Res	Res	Res	Res	Res	Res	PLLSRC
															RW

Bit	Name	R/W	Reset Value	Function
31:1	Reserved			
0	PLLSRC	RW	0	PLL clock source selection: 0: HSI 1: HSE

8.6.5. External clock source control register (RCC_ECSCR)

Address offset: 0x10

Reset value: 0x0002 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		LSE_DRIVER	
														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	Res	Res	Res	HSE_FREQ		Res	
												RW	RW		

Bit	Name	R/W	Reset Value	Function
31:18	Reserved	RES	-	Resserved
17:16	LSE_DRIVER	RW	0x10	Low-speed crystal oscillator drive capability selection. 00: close LSE 01: Weak driving ability 10: Default drive capability (recommended) 11: The strongest driving ability Note: The proper drive capability needs to be selected according to the crystal characteristics, load capacitance and parasitic parameters of the circuit board. The greater the driving ability, the greater the power consumption and the weaker the driving ability, the less the power consumption.
15:4	Reserved		-	
3:2	HSE_FREQ	RW	0x0	HSE crystal oscillator operating frequency. 00: HSE off 01: 4 MHz to 8 MHz 10: 8 MHz to 16 MHz 11: 16 MHz to 32 MHz
1:0	Resserved			

8.6.6. Clock interrupt enable register (RCC_CIER)

Address offset: 0x18

Reset value: 0x0000 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	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	Res	Res	Res	PLL RDYIE	HSE RDYIE	HSI RDYIE	Res	LSE RDYIE	LSI RDYIE
										RW	RW	RW		RW	RW

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	PLLRDYIE	RW	0	PLL ready interrupt enable. 0: Disable 1: Enable
4	HSERDYIE	RW	0	HSE clock ready interrupt enable. 0: Disable 1: Enable
3	HSIRDYIE	RW	0	HSI clock ready interrupt enable. 0: Disable 1: Enable
2	Reserved	-	-	Reserved
1	LSERDYIE	RW	0	LSE clock ready interrupt enable. 0: Disable 1: Enable
0	LSIRDYIE	RW	0	LSI clock ready interrupt enable. 0: Disable 1: Enable

8.6.7. Clock interrupt flag register (RCC_CIFR)

Address offset: 0x1C

Reset value: 0x0000 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	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	LSE CSSF	CSSF	Res	Res	PLL RDYF	HSE RDYF	HSI RDYF	Res	LSE RDYF	LSI RDYF
						R	R			R	R	R		R	R

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	LSECSSF	R	0	LSE clock security system (CSS) interrupt flag. When hardware detects LSE, this register is set when the OSC clock fails. 0: LSE clock detection failure interrupt is not generated, 1: LSE clock detection failure interrupt generation, Programming LSECSSC register 1 clears this bit.
8	CSSF	R	0	HSE clock security system interrupt flag. When hardware detects LSE, this register is set when the OSC clock fails. 0: HSE clock detection failure interrupt is not generated, 1: HSE clock detection failure interrupt generation, Programming CSSC register 1 clears this bit.
7:6	Reserved	-	-	Reserved
5	PLLRDYF	R	0	PLL ready interrupt flag bit When PLL lock and PLLRDYIE bit is set. Software clears this bit by setting the PLLRDYC bit. 0: No clock ready interrupt caused by PLL lock 1: Clock ready interrupt caused by PLL lock
4	HSERDYF	R	0	HSE ready interrupt flag This bit is set by hardware when HSE is stable and HSERDYIE is enabled. Software clears this bit by setting the HSERDYC bit. 0: No clock ready interrupt caused by HSE 1: Clock ready interrupt caused by HSE
3	HSIRDYF	R	0	HIS ready interrupt flag This bit is set by hardware when HSI is stable and HSIRDYIE is enabled. Software clears this bit by setting the HSIRDYC bit. 0: No clock ready interrupt caused by HSI 1: Clock ready interrupt caused by HSI
2	Res	-	-	Reserved
1	LSERDYF	R	0	LSE ready interrupt flag This bit is set by hardware when LSE is stable and LSE RDYIE is enabled. Software clears this bit by setting the LSERDYC bit. 0: No clock ready interrupt caused by LSE 1: Clock ready interrupt caused by LSE
0	LSIRDYF	R	0	LSI ready interrupt flag This bit is set by hardware when LSI is stable and LSIRDYIE is enabled. Software clears this bit by setting the LSIRDYC bit. 0: No clock ready interrupt caused by LSI 1: Clock ready interrupt caused by LSI

8.6.8. Clock interrupt clear register (RCC_CICR)

Address offset: 0x20

Reset value: 0x0000 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	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	LSECSSC	CSSC	Res	Res	PLLRDYC	HSERDYC	HSIRDYC	Res	LSERDYC	LSIRDYC
						W	W			W	W	W		W	W

Bit	Name	R/W	Reset Value	Function
31:10	Reserved	-	-	Reserved
9	LSECSSC	W	0	LSE clock security system (CSS) interrupt flag is cleared. 0: No effect, 1: Clear the LSECSSF flag
8	CSSC	W	0	Clock safe interrupt clear bit. 0: No effect. 1: Clear the CSSF flag.
7:6	Reserved	-	-	Reserved
5	P LLRDYC	W	0	PLL ready flag is cleared. 0: No effect. 1: Clear the PLLRDYF bit.
4	HSERDYC	W	0	HSE ready flag is cleared. 0: No effect. 1: Clear the HSERDYF bit.
3	HSIRDYC	W	0	HSI ready flag is cleared. 0: No effect. 1: Clear the HSIRDYF bit.
2	Reserved	-	-	Reserved
1	LSERDYC	W	0	LSE ready flag is cleared. 0: No effect. 1: Clear the LSERDYF bit.
0	LSIRDYC	W	0	LSI ready flag is cleared. 0: No effect. 1: Clear the LSIRDYF bit.

8.6.9. I/O interface reset register (RCC_IOPRSTR)

Address offset: 0x24

Reset value: 0x0000 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	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	Res	Res	Res	GPIOFRST	Res	Res	Res	GPIOBRST	GPIOARST
										RW				RW	RW

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	GPIOFRST	RW	0	I/O PortF reset. 0: no effect, 1: PortF I/O reset
4:2	Reserved	-	-	Reserved
1	GPIOBRST	RW	0	I/O PortB resets. 0: no effect, 1: Port B I/O reset
0	GPIOARST	RW	0	I/O PortA resets. 0: no effect, 1: PortA I/O reset

8.6.10. AHB peripheral reset register (RCC_AHBRSTR)

Address offset: 0x28

Reset value: 0x0000 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	Res	Res	Res

15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	CRC RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	DMA RST
			RW												RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12	CRCRST	RW	0	CRC module reset. 0: no effect, 1: CRC module reset,
11:9	Reserved	-	-	Reserved
8:1	Reserved	-	-	Reserved
0	DMARST	RW	0	DMA reset. 0: no effect, 1: DMA module reset

8.6.11. APB peripheral reset register 1 (RCC_APBRSTR1)

Address offset: 0x2C

Reset value: 0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LPTIM RST	Res	Res	PWR RST	DBG RST	Res	Res	Res	Res	Res	I2C RST	Res	Res	Res	USART2 RST	Res
RW			RW	RW						RW				RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	SPI2 RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	TIM3 RST	Res
	RW													RW	

Bit	Name	R/W	Reset Value	Function
31	LPTIMRST	RW	0	LP Timer module reset. 0: no effect, 1: The module is reset,
30:29	Reserved	-	-	Reserved
28	PWRRST	RW	0	Power interface module reset. 0: no effect, 1: The module is reset,
27	DBGRST	RW	0	MCU Debug module reset. 0: no effect, 1: The module is reset,
26:22	Reserved	-	-	Reserved
21	I2CRST	RW	0	I2C1 module reset. 0: no effect, 1: The module is reset,
20:18	Reserved	-	-	Reserved
17	USART2RST	RW	0	USART2 module reset. 0: no effect, 1: The module is reset
16:15	Reserved	-	-	Reserved
14	SPI2RST	RW	0	SPI2 module reset. 0: no effect, 1: The module is reset,
13:2	Reserved	-	-	Reserved
1	TIM3RST	RW	0	TIM3 module reset. 0: no effect, 1: The module is reset
0	Reserved			

8.6.12. APB peripheral reset register 2 (RCC_APBRSTR2)

Address offset: 0x30

Reset value: 0x0000 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	LED RST	COMP2 RST	COMP1 RST	ADC RST	Res	TIM17 RST	TIM16 RST	Res
								RW	RW	RW	RW		RW	RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIM14 RST	USART1 RST	Res	SPI1 RST	TIM1 RST	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SYSCFG RST
RW	RW		RW	RW											RW

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	LEDRST	RW	0	LED module reset. 0: no effect, 1:The module is reset,
22	COMP2RST	RW	0	COMP2 module reset. 0: no effect, 1: The module is reset,
21	COMP1RST	RW	0	COMP1 module reset. 0: no effect, 1: The module is reset,
20	ADCRST	RW	0	ADC module reset. 0: no effect, 1: The module is reset,
19	Reserved	-	-	Reserved
18	TIM17RST	RW	0	TIM17 module reset. 0: no effect, 1: The module is reset,
17	TIM16RST	RW	0	TIM16 module reset. 0: no effect, 1: The module is reset,
16	Reserved	-	-	Reserved
15	TIM14RST	RW	0	TIM14 module reset. 0: no effect, 1: The module is reset,
14	USART1RST	RW	0	USART1 module reset. 0: no effect, 1: The module is reset,
13	Reserved	-	-	Reserved
12	SPI1RST	RW	0	SPI1 module reset. 0: no effect, 1: The module is reset,
11	TIM1RST	RW	0	TIM1 module reset. 0: no effect, 1: The module is reset,
10:1	Reserved	-	-	Reserved
0	SYSCFGRST	RWs	0	SYSCFG module reset. 0: no effect, 1: The module is reset,

8.6.13. I/O interface clock enable register (RCC_IOPENR)

Address offset: 0x34

Reset value: 0x0000 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	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	Res	Res	Res	GPIOFEN	Res	Res	Res	GPIOBEN	GPIOAEN
										RW				RW	RW

Bit	Name	R/W	Reset Value	Function
31:6	Reserved	-	-	Reserved
5	GPIOFEN	RW	0	I/O PortF clock enable. 0: Clock disabled, 1: Clock enable
4:2	Reserved	-	-	Reserved
1	GPIOBEN	RW	0	I/O PortB clock enable. 0: Clock disabled, 1: Clock enable
0	GPIOAEN	RW	0	I/O PortA clock enable. 0: Clock disabled, 1: Clock enable

8.6.14. AHB peripheral clock enable register (RCC_AHBENR)

Address offset: 0x38

Reset value: 0x0000 0300

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	CRC EN	Res	Res	Res	FLASH EN	Res	Res	Res	Res	Res	Res	Res	DMA EN
			RW				RW								RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved	-	-	Reserved
12	CRCEN	RW	0	CRC module clock enable. 0: Disable 1: Enable
11:10	Reserved	-	-	Reserved
9	SRAMEN	RW	1	In sleep mode, the clock enable control of SRAM 0: The module clock is disabled in sleep mode 1: The module clock is enabled in sleep mode Note: This bit only affects the clock enable of this module in sleep mode, in run mode, the clock of this module will not be disabled
8	FLASHEN	RW	1	In sleep mode, the clock enable control of FLASH 0: The module clock is disabled in sleep mode 1: The module clock is enabled in sleep mode Note: This bit only affects the clock enable of this module in sleep mode, in run mode, the clock of this module will not be disabled
7:1	Reserved	-	-	Reserved
0	DMAEN	RW	0	DMA module clock enable. 0: Disable 1: Enable

8.6.15. APB peripheral clock enable register 1 (RCC_APBENR1)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM EN	Res	Res	PWR EN	DBG EN	Res	Res	Res	Res	Res	I2C EN	Res	Res	Res	USART2 EN	Res
RW			RW	RW						RW				RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	SPI2EN	Res	Res	WWDG EN	RTCAPB EN	Res	Res	Res	Res	Res	Res	Res	Res	TIM3 EN	Res
	RW			RW	RW		RW							RW	

Bit	Name	R/W	Reset Value	Function
31	LTIMEN	RW	0	LP Timer1 module clock enable. 0: Disable 1: Enable
30:29	Reserved	-	-	Reserved
28	PWREN	RW	0	Low power control block clock enable. 0: Disable 1: Enable
27	DBGEN	RW	0	Debug module clock enable. 0: Disable 1: Enable
26:22	Reserved	-	-	Reserved
21	I2CEN	RW	0	I2C1 module clock enable. 0: Disable 1: Enable
20:18	Reserved	-	-	Reserved
17	USART2EN	RW	0	USART2 module clock enable. 0: Disable 1: Enable
16:15	Reserved	-	-	Reserved
14	SPI2EN	RW	0	SPI2 module clock enable. 0: Disable 1: Enable
13:12	Reserved	-	-	Reserved
11	WWDGEN	RW	0	Window WDG module clock enable. 0: Disable 1: Enable This register is cleared by hardware system reset.
10	RTCAPBEN	RW	0	RTC Module APB clock enable. 0: Disable 1: Enable
9:2	Reserved	-	-	Reserved
1	TIM3EN	RW	0	TIM3 module clock enable. 0: Disable 1: Enable
0	Reserved			

8.6.16. APB peripheral clock enable register 2 (RCC_APBENR2)

Address offset: 0x40

Reset value: 0x0000 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	LEDEN	COMP2EN	COMP1EN	ADCEN	Res	TIM17EN	TIM16EN	Res
								RW	RW	RW	RW		RW	RW	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIM14EN	USART1EN	Res	SPI1EN	TIM1EN	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SYSCFGEN
RW	RW		RW	RW											RW

Bit	Name	R/W	Reset Value	Function
31:24	Reserved	-	-	Reserved
23	LEDEN	RW	0	LED module clock enable. 0: Disable 1: Enable
22	COMP2EN	RW	0	COMP2 module clock enable. 0: Disable 1: Enable
21	COMP1EN	RW	0	COMP1 module clock enable. 0: Disable 1: Enable
20	ADCEN	RW	0	ADC module clock enable. 0: Disable

				1: Enable
19	Reserved	-	-	Reserved
18	TIM17EN	RW	0	TIM17 module clock enable. 0: Disable 1: Enable
17	TIM16EN	RW	0	TIM16 module clock enable. 0: Disable 1: Enable
16	Reserved	-	-	Reserved
15	TIM14EN	RW	0	TIM14 module clock enable. 0: Disable 1: Enable
14	USART1EN	RW	0	USART1 module clock enable. 0: Disable 1: Enable
13	Reserved	-	-	Reserved
12	SPIEN	RW	0	SPI1 module clock enable. 0: Disable 1: Enable
11	TIM1EN	RW	0	TIM1 module clock enable. 0: Disable 1: Enable
10:1	Reserved	-	-	Reserved
0	SYSCFGEN	RW	0	SYSCFG module clock enable. 0: Disable 1: Enable

8.6.17. Peripheral independent clock configuration register (RCC_CCIPR)

Address offset: 0x54

Reset value: 0x0000 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	LPTIM1SEL[1:0]		Res	Res
												RW	RW		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res		Res		Res		COMP2SEL	COMP1SEL	PVDSEL	Res	Res	Res	Res	Res	Res	Res
						RW	RW	RW							

Bit	Name	R/W	Reset Value	Function
31:20	Reserved	-	-	Reserved
19 :18	LPTIMSEL[1:0]	RW	2'b00	LPTIM1 internal clock source selection. 00: PCLK 01: LSI 10: No clock 11: LSE
17:10	Reserved	-	-	Reserved
9	COMP2SEL	RW	0	COMP2 module clock source selection. 0: PCLK 1: LSC (Clock selected by RCC_BDCR.LSCOSEL) Note: Configure the selected LSC clock before enabling FLTEN.
8	COMP1SEL	RW	0	COMP1 module clock source selection. 0: PCLK 1: LSC (Clock selected by RCC_BDCR.LSCOSEL) Configure this register to select the clock before enabling COMP2_FR2.FLTEN.
7	PVDSEL	RW	0	PVD detect clock source selection. 0: PCLK 1: LSC (Clock selected by RCC_BDCR.LSCOSEL) Note:Configure this register to select the clock before enabling COMP1_FR1.FLTEN.
6:0	Reserved	-	-	Reserved

8.6.18. RTC domain control register (RCC_BDCR)

Address offset: 0x5C

Reset value: 0x0000 0000, reset by POR/BOR

When PWR_CR1.DBP is 1, it is allowed to write to this register.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	LSCOSEL	LSCOEN	Res	Res	Res	Res	Res	Res	Res	BDRST
						RW	RW								RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RTCEN	Res	Res	Res	Res	Res	RTCSEL[1:0]		Res	LSECSSD	LSECSSON	Res		LSEBYP	LSERDY	LSEON
RW						RW			RW	RW			RW	R	RW

Bit	Name	R/W	Reset Value	Function
31:26	Reserved	-	-	Reserved
25	LSCOSEL	RW	0	Low-speed clock selection. 0: LSI 1: LSE
24	LSCOEN	RW	0	Low-speed clock enable. 0: Disable 1: Enable
23:17	Reserved	-	-	Reserved
16	BDRST	RW	0	RTC domain soft reset. 0: No effect 1: Reset
15	RTCEN			RTC clock enable. Software set or reset. 0: Disable 1: Enable
14:10	Reserved	-	-	Reserved
9:8	RTCSEL[1:0]	RW	0	RTC clock source selection. 00: No clock 01: LSE 10: LSI 11: HSE divided by 128 Once the RTC clock source is selected, it cannot be changed, except in the following cases: ● RTC is reset to 00 ● Selected as LSE (LSECSSD = 1) but no LSE ● BDRST soft reset to 00
7	Reserved	-	-	Reserved
6	LSECSSD	R	0	LSE CSS (clock security system) detection failed. This bit is set by hardware to indicate that CSS detects 32.768 kHz OSC (LSE) failed. 0: No LSE detected failure 1: Failed to detect ISE
5	LSECSSON	RW	0	LSE CSS enabled 0: Disable 1: Enable LSECSSON must be enabled after LSEON = 1 and LSERDY = 1. Once this bit is enabled, it cannot be disabled unless LSECSSD = 1.
4:3	Reserved	-	-	-
2	LSEBYP	RW	0	LSE OSC bypass 0: Not bypassed, the low-speed external clock selects the crystal oscillator 1: Bypassed, the low-speed external clock selects the external interface input clock Note: This bit can only be written when the external 32.768 kHz OSC is disabled (LSEON = 0 and LSERDY = 0).
1	LSERDY	R	0	LSE OSC ready bit. Set by hardware, cleared by hardware, indicating when LSE is stable

				0: Not ready 1: Ready
0	LSEON	RW	0	LSE OSC enabled. 0: Disable 1: Enable

8.6.19. Control/status register (RCC_CSR)

Address offset: 0x60

Reset value: 0x0000 0000

Reset sources: 1) [30:25]: POR reset, 2) LSION: system reset, 3) NRST_FLTIDS will not be reset by system reset reset

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	WWDG RSTF	IWDG RSTF	SFTRSTF	PWRRSTF	PINRSTF	OBLRSTF	Res	RMVF	Res	Res	Res	Res	Res	Res	Res
	R	R	R	R	R	R		RW							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	Res	NRST_FLT-DIS	Res	Res	Res	Res	Res	Res	LSIRDY	LSION
							RW							R	RW

Bit	Name	R/W	Reset Value	Function
31	Reserved			
30	WWDGRSTF	R	0	Window WDG reset flag. Setting RMVF to 1 clears this bit.
29	IWDGRSTF	R	0	IWDG reset flag. Setting RMVF to 1 clears this bit.
28	SFTRSTF	R	0	Soft reset flag. Setting RMVF to 1 clears this bit.
27	PWRRSTF	R	0	BOR/POR/PDR reset flag. Setting RMVF to 1 clears this bit.
26	PINRSTF	R	0	External NRST pin reset flag. Setting RMVF to 1 clears this bit.
25	OBLRSTF	R	0	Option byte loader reset flag. Setting RMVF to 1 clears this bit.
24	Reserved			-
23	RMVF	RW	0	Reset flags [30:25] need to be cleared by software.
8	NRST_FLTDIS	RW	0	NRST filter disabled 0: HSI_10M is enabled, and the filter 20 us width function is enabled 1: The filtering function is disabled, and HSI_10M remains off
7:2	Reserved	-	-	Reserved
1	LSIRDY	R	0	LSI OSC stable flag. 0: LSI is not stable 1: LSI has stabilized
0	LSION	RW	0	LSI OSC enabled. 0: Disable 1: Enable Set by software, cleared by software. This bit is set by hardware when IWDG is enabled by hardware (via option byte) and LSECSS is enabled by software.

8.6.20. RCC register address map

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

0x00	RCC_CR	Res.	Res.	Res.	Res.	Res.	Res.	PLLRDY	PLLON	Res.	Res.	Res.	Res.	HSE_CSS	HSEBYP	HSERDY	HSEON	Res.	Res.	HSIDIV [2:0]			HSIRDY	Res.	HSION	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value							0	0					0	0	0	0			0	0	0	0	0	1								
0x04	RCC_ICSCR	Res.	Res.	Res.	Res.	LSI_STARTUP[1:0]		Res.	LSI_TRIM[8:0]									HSI_FS[2:0]			HSI_TRIM[12:0]												
	Reset value					0	0		0	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	1
0x08	RCC_CFGR	Res.	MCOPRE[2:0]			Res.	MCOSEL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PPRE [2:0]			HPRE[3:0]				Res.	Res.	SWS[2:0]			SW[2:0]		
	Reset value		0	0	0		0	0	0										0	0	0	0	0	0	0			0	0	0	0	0	0
0x0C	RCC_PLLCFGR	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.	PLLSRC
	Reset value																																0
0x10	RCC_ECSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE_DRIVER[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSE_FREQ[1:0]		Res.	Res.
	Reset value															1	0													0	0		
0x14	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.
0x18	RCC_CIER	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.	PLLRDYI	HSER-	HSIRDYI	Res.	LSER-	LSIRDYI
	Reset value																											0	0	0		0	0
0x1C	RCC_CIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSECSS	Res.	Res.	Res.	PLLRDYI	HSER-	HSIRDYI	Res.	LSER-	LSIRDYI
	Reset value																							0				0	0	0		0	0
0x20	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSECSS	Res.	Res.	Res.	PLLRDYI	HSER-	HSIRDYI	Res.	LSER-	LSIRDYI
	Reset value																							0				0	0	0		0	0
0x24	RCC_IOPRSTR	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.	GPIOFRS	Res.	Res.	Res.	GPIOBRS	GPIOARS

0x44-0x50	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.
0x40	RCC_APBENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEDEN	COMP2EN	COMP1EN	ADCEN	Res.	TIM17EN	TIM16EN	Res.	TIM14EN	USART1EN	Res.	SPI1EN	TIM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value										0	0	0	0		0	0		0	0		0	0								0
0x3C	RCC_APBENR1	LPTIMEN	Res.	Res.	PWREN	DBGEN	Res.	Res.	Res.	Res.	Res.	Res.	I2CEN	Res.	Res.	Res.	USART2EN	Res.	Res.	SPI2EN	Res.	Res.	WWDGEN	RTCAPBEN	Res.	Res.	Res.	Res.	Res.	TIM3EN	Res.
	Reset value	0			0	0							0				0			0			0	0						0	
0x38	RCC_AHBENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	Res.	SRAMEN	FLASHE	Res.	Res.	Res.	Res.	DMAEN
	Reset value																					0			1	1					0
0x34	RCC_IOPENR	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.	GPIOBEN	GPIOAEN
	Reset value																													0	0
0x30	RCC_APBRSTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEDRST	COMP2RST	COMP1RST	ADCRST	Res.	TIM17RST	TIM16RST	Res.	TIM14RST	USART1RST	Res.	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
	Reset value										0	0	0			0	0		0	0		0	0								0
0x2C	RCC_APBRSTR1	LPTIMRST	Res.	Res.	PWRRST	DBGRST	Res.	Res.	Res.	Res.	Res.	Res.	I2CRST	Res.	Res.	Res.	USART2RST	Res.	Res.	SPI2RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM3RST	Res.
	Reset value				0	0							0				0			0									0		
0x28	RCC_AHBRSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMARST
	Reset value																					0									0
	Reset value																													0	0

0x54	RCC_CCIPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIMSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP2SEL	COMP1SEL	PVDSEL	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0	0								0	0	0						
0x58	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.
0x5C	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	Res.	Res.	LSEBYP	LSERDY	LSEON
	Reset value							0	0								0	0						0	0		0	0			0	0	0
0x60	RCC_CSR	Res.	WWD-RSTF	IWDGRSTF	SFTRSTF	PWRRSTF	PINRSTF	OBLRSTF	Res.	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NRST_FLT-	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value		0	0	0	0	0	0		0															0						0	0

9. General-purpose I/Os (GPIO)

9.1. GPIO introduction

Each GPIO port has:

Four 32-bit configuration registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR)

Two 32-bit data registers (GPIOx_IDR and GPIOx_ODR)

One 32-bit set/reset register (GPIOx_BSRR)

One 32-bit lock register (GPIOx_LCKR)

Two alternate function selection registers (GPIOx_AFRH and GPIOx_AFRL).

9.2. GPIO main features

- Output status: push-pull or open drain + pull-up/down
- Output data from output data register (GPIOx_ODR) or peripheral (alternate function output)
- Speed selection for each I/O
- Input status: floating, pull-up/down, analog
- Input data to input data register (GPIOx_IDR) or peripheral (alternate function input)
- Bit set/reset register (GPIOx_BSRR) for bitwise write access to GPIOx_ODR
- Locking mechanism (GPIOx_LCKR) provided to freeze the I/O port configuration function
- Analog function
- Alternate functions selection registers (at most 16 AFs per I/O port)
- Fast toggle capable of changing every single cycle
- Highly flexible pin multiplexing allows the use of I/O pins as GPIOs or as one of several peripheral function

9.3. GPIO functional description

Each port bit of the GPIO ports can be individually configured by software in several modes:

- Input floating
- Input pull-up
- Input pull-down
- Analog input
- Output open-drain with pull-up or pull-down capability
- Output puss-pull with pull-up or pull -down capability
- Alternate function push-pull with pull-up or pull–down capability
- Alternate function open-drain with pull-up or pull-down capability

Each I/O port bit is freely programmable, however the I/O port registers have to be accessed as 32-bit words, half-words or bytes. The purpose of the GPIOx_BSRR and GPIOx_BRR registers is to allow atomic read/modify accesses to any of the GPIOx_ODR registers. In this way, there is no risk of an IRQ occurring between read and modify access.

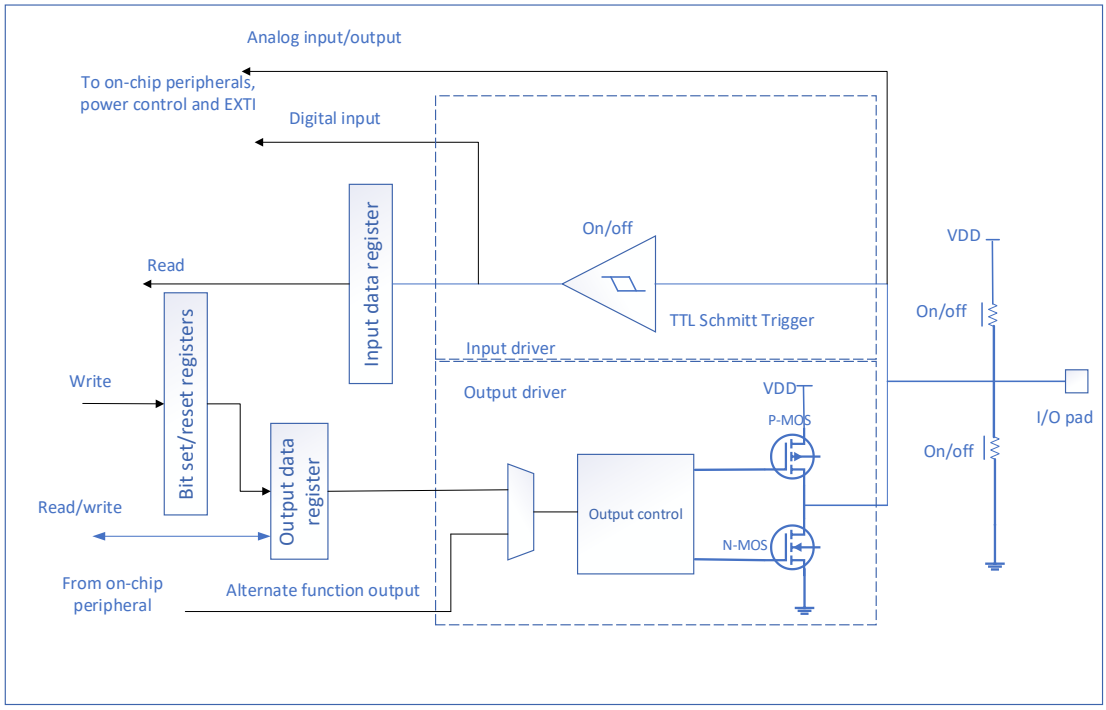


Figure 9-1 Basic structure of an I/O port bit

9.3.1. General-purpose I/O (GPIO)

During and after reset, the alternate functions are not active and most of the IOs are configured in analog mode.

The debug pins are in alternate function pull-up or pull-down after reset:

- PA14-SWCLK: in pull-down mode
- PA13-SWDIO: in pull-up mode

Boot pin is set to input pull-down mode after reset:

- PF4-Boot: in pull-down mode

When the pin is configured as output, the value written to the output data register (GPIOx_ODR) is output on the I/O pin. It is possible to use the output drive in push-pull mode or open-drain mode (only the low level is driven, high level is HI-Z).

The input data register (GPIOx_IDR) captures the data present on the I/O pin at every AHB clock cycle.

All GPIO pins have weak internal pull-up and pull-down resistors, which can be activated or not depending on the value in the GPIOx_PUPDR register.

9.3.2. I/O pin alternate function multiplexer and mapping

The device I/O pins are connected to on-board peripherals/modules through multiplexers that allows only one peripheral alternate function (AF) connected to an I/O pin at a time. In this way, there can be no conflict between peripherals available on the same I/O pin.

Each I/O port has a multiplexer with up to 16 alternate function inputs (AF0 to AF7), which can be configured through the registers GPIOx_AFRL (for pin 0 to 7) and GPIOx_AFRH (for pin 8 to 15).

- After reset, the multiplexer selection is AF0. The I/Os are configured in alternate function mode through GPIOx_MODER register.
- The alternate function assignments for each pin are details in section 2.3.

In addition to this flexible multiplexer architecture, each peripheral has alternate functions mapped onto different I/O pins to optimize the number of peripherals available in smaller packages.

The user configures IO as follows:

- Debug function: After each reset, these pins are assigned as alternate function pins immediately usable by the debugger host.
- GPIO: Configure the corresponding I/O port as output, input or analog mode in GPIOx_MODER register.
- Peripheral multiplexing function:
  - The I/O corresponding to the register GPIOx_AFRL or GPIOx_AFRH configuration is the alternate function x (x = 0… 15).
  - Registers GPIOx_OTYPER, GPIOx_PUPDR and GPIOX_OSPEEDER configure the type, pull-up/pull-down and output speed respectively.
  - Configure the corresponding I/O as an alternate function in the GPIOx_MODER register.
- Additional functions:
  - ADC and COMP functions are enabled in the registers of the ADC and COMP modules, in every I/O configuration. When the I/O is used as ADC or COMP, it is recommended to configure the port as analog mode through the register GPIOx_MODER
  - For additional functions of the crystal oscillator, configure the respective functions in the corresponding PWR and RCC module registers. These configurations have higher priority than standard GPIO configurations.

9.3.3. I/O port control registers

Each of the GPIO ports has four 32-bit memory-mapped control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR and GPIOx_PUPDR) to configure up to 16 I/Os. The register GPIOx_MODER is used to select the I/O mode (input, output, AF, analog). The GPIOx_OTYPER and GPIOx_OSPEEDR registers are used to select the output type (push-pull or open-drain) and speed. The GPIOx_PUPDR register is used to select the pull-up/pull-down whateverr the I/O direction.

9.3.4. I/O port data registers

Each GPIO has two 16-bit memory-mapped data registers: input and output data registers (GPIOx_IDR and GPIOx_ODR). GPIOx_ODR stores the data to be output, it is read/written accessible. The data input through the I/O are stored into the input data register (GPIOx_IDR), a read-only register.

9.3.5. I/O data bitwise handling

The bit set reset register (GPIOx_BSRR) is a 32-bit register that allows the application to set and reset each individual bits in the output data register (GPIOx_ODR). The bit set reset register has twice the size of GPIOx_ODR.

To each bit in GPIOx_ODR, correspond two control bits of GPIOx_BSRR: BS(i) and BR(i). When written bit BS(i) to 1 can set the corresponding bit of GPIOx_ODR to 1, and setting bit BR(i) to 1 can clear the corresponding bit of GPIOx_ODR to 0.

Write any bit to 0 in GPIOx_BSRR does not have any effect on the corresponding bit in GPIOx_ODR. If there is an attempt to both set and reset a bit in GPIOx_BSRR, the set operation has priority.

Using the GPIOx_BSRR register to change the values of individual bit in GPIOx_ODR is a “one-shot” effect that does not lock the GPIOx_ODR bits. The GPIOx_ODR bits can always be accessed directly. The GPIOx_BSRR register provides a way of performing atomic bitwise handling.

There is no need for the software to disable interrupts when programming the GPIOx_ODR at bit level: it is possible to modify one or more bits in a sinle atomic AHB write access.

9.3.6. GPIO locking mechanism

It is possible to freeze the IO control with GPIOx_LCKR registers through a series of special write timings, including GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH.

A special write/read sequence can manipulate the register GPIOx_LCKR. When the right lock sequence is applied to bit 16 in this register, the value of LCKR[15:0] can LOCK the I/O (during the write sequence, the value of LCKR[15:0] remains unchanged). When the LOCK sequence has been applied to a port bit, the value of the port bit cannot be modify until the next MCU reset or peripheral reset. Each GPIOx_LCKR bit freezes the corresponding bit in the control registers (GPIOx_MODER, GPIOx_OTYPER, GPIOx_OSPEEDR, GPIOx_PUPDR, GPIOx_AFRL and GPIOx_AFRH).

the GPIOx_LCKR register with a word (32 bits) because the [15:0] bits are also set when the GPIOx_LCKR bit 16 is set.

9.3.7. I/O alternate function input/output

Two registers are provided to select one of the alternate function input/outputs available for each I/O. The user can connect an alternate function to the IO port according as required by the application.

This means that a number of possible peripheral functions are multiplexed on each GPIO using the GPIOx_AFRL and GPIOx_AFRH alternate function registers. The application can thus select any one of the possible functions for each I/O. The AF selection signal being common to the alternate function input and alternate function output, a single channel is selected for the alternate function input/output of a given I/O.

9.3.8. External interrupt/wakeup lines

All ports have external interrupt capability. To use the external interrupt lines, the given pin must be disabled in analog mode or as oscillator pin, so the input trigger is kepy enabled.

9.3.9. I/O input configuration

When the I/O port is configured as input:

- The output buffer is disabled
- The Schmitt trigger input is enable
- The pull-up and pull-down resistors can be enabled/disabled according to the configuration of the GPIOx_PUPDR register
- The data present on the I/O pins are sampled into the input data register on every AHB clock cycle
- A read access to the input data register provides the I/O status

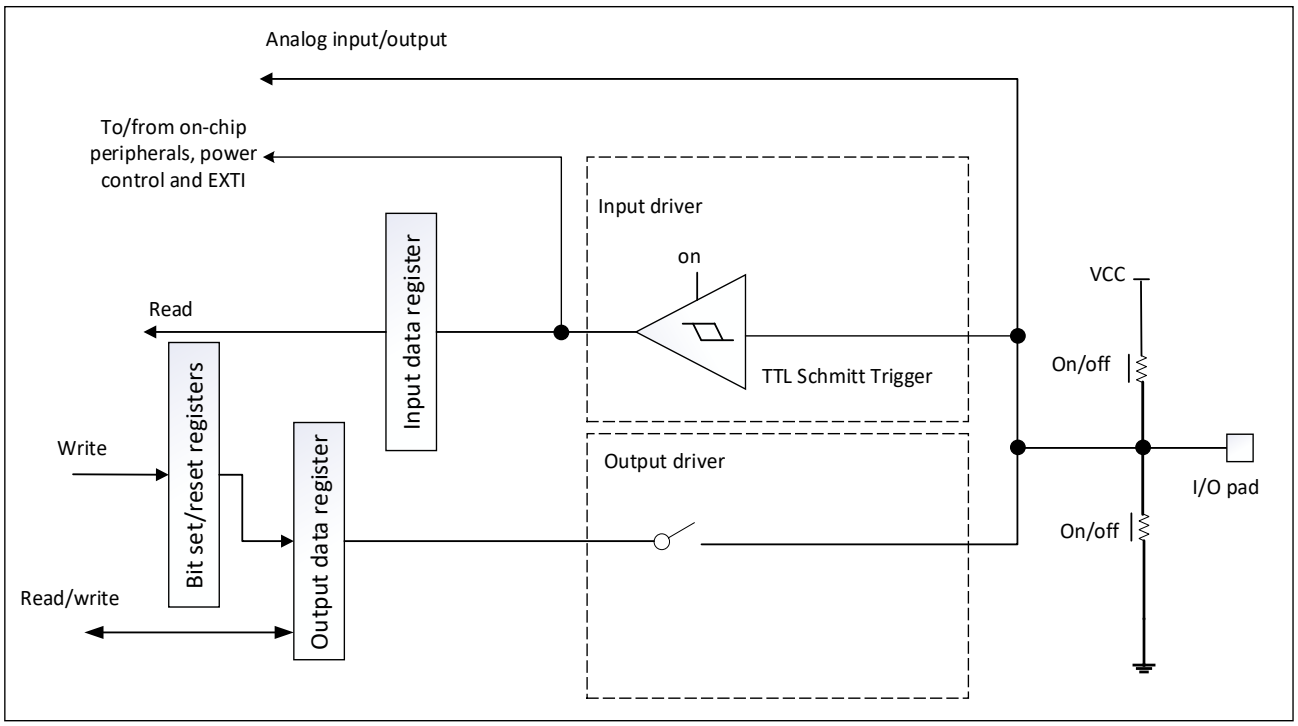


Figure 9-2Input floating/pull up/pull down configurations

9.3.10. I/O output configuration

When the I/O port is configured as output:

- The output buffer is enabled:
  - Open-drain mode: A '0' in the output register activates the N-MOS whereas a '1' in the output register leaves the port in a high-impedance state (the PMOS is never activated).
  - Push-pull mode: A '0' in the output register activates the N-MOS whereas a '1' in the output register activates the P-MOS.
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors can be enabled/disabled according to the configuration of the GPIOx_PUPDR register
- The data present on the I/O pins are sampled into the input data register every AHB clock cycle
- A read access to the input data register gets the I/O state
- A read access to the output data register gets the value of the last write

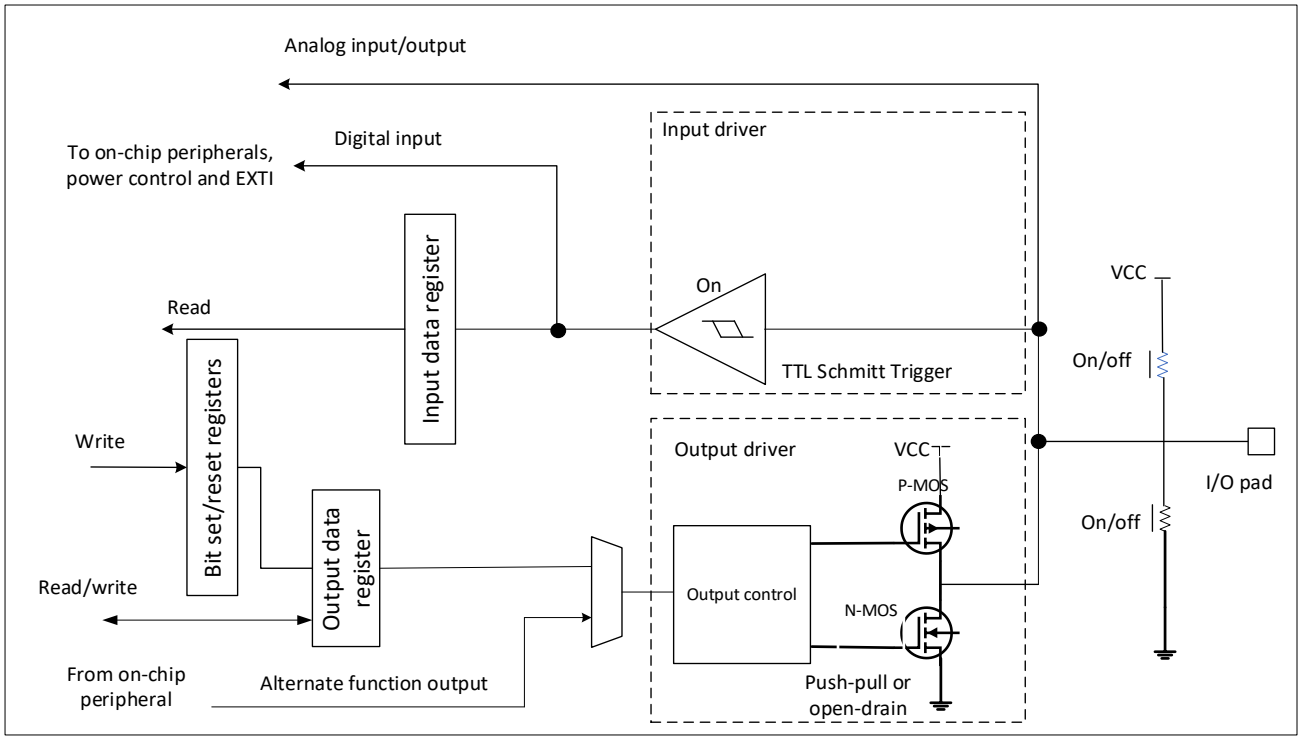


Figure 9-3 Output configuration

9.3.11. Alternate function configuration

When an I/O port is configured as alternate function:

- In an open-drain or push-pull configuration, the output buffer is turned on
- Built-in peripheral signal-driven output buffer (multiplexed function output)
- The Schmitt trigger input is activated
- The pull-up and pull-down resistors can be enabled/disabled according to the configuration of the GPIOx_PUPDR register
- The data present on the I/O pins are sampled into the input data register every AHB clock cycle
- A read access to the input data register gets the I/O state

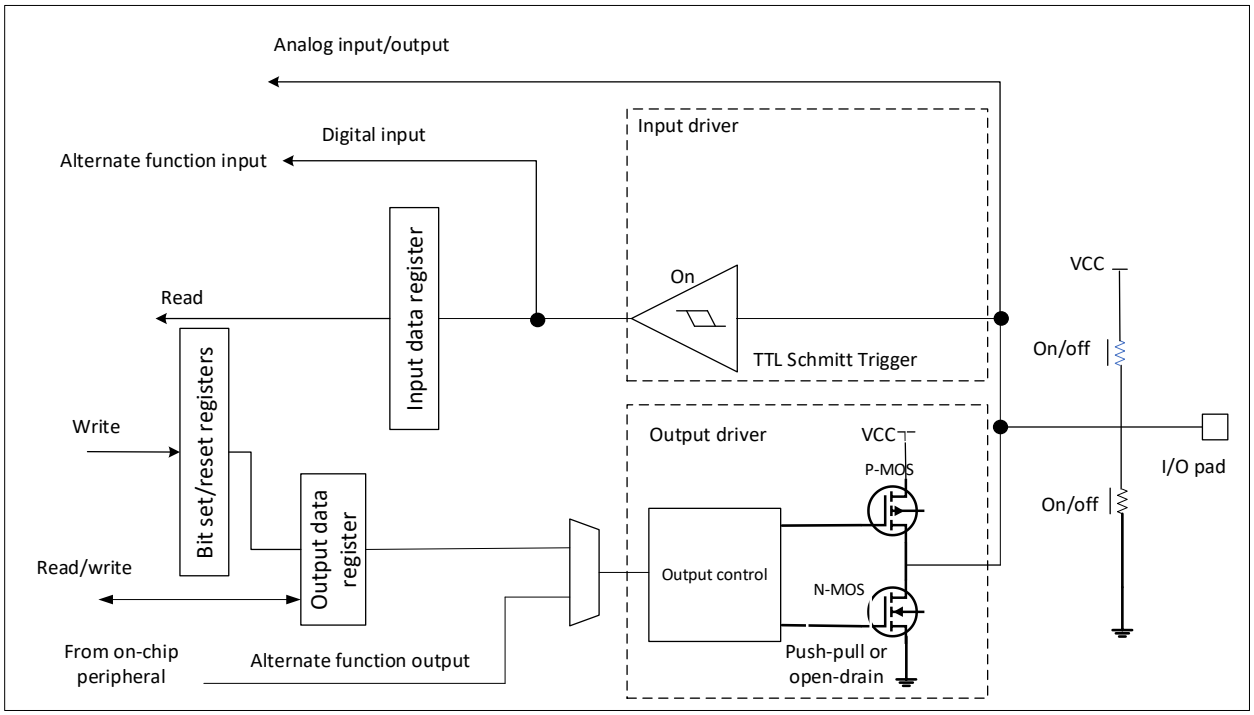


Figure 9-4 Alternate function configuration

9.3.12. Analog configuration

When an I/O port is configured as analog configuration:

- The output buffer is disabled
- The Schmitt trigger input is deactivated, providing zero consumption for every analog value of the I/O pin. The output of Schmitt trigger is forced to '0'
- The weak pull-up and pull-down resistors are disabled (software setting required)
- Read access to the input data register gets the value is '0'

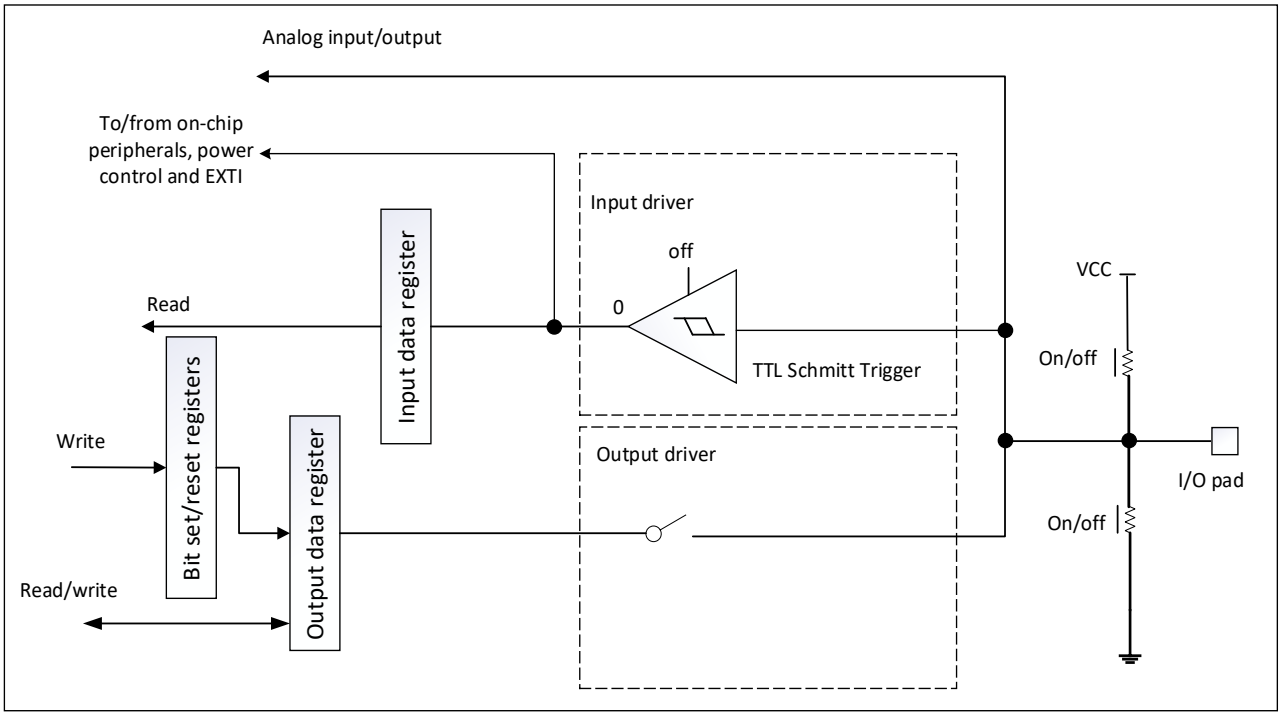


Figure 9-5 High impedance-analog configuration

9.3.13. Use the HSE/LSE oscillator pins as GPIOs

When the HSE or LSE oscillator is switch off (default state after reset), the related oscillator pins can be used as normal GPIOs.

When the HSE or LSE oscillator is switch on (by setting the HSEON or LSEON bit in the RCC_CSR register) the corresponding port needs to be configured as an analog port by software.

When the crystal oscillator is configured in a user external clock mode, only the pin is reserved for clock input and the OSC_IN or OSC32_IN pin can still be used as normal GPIO.

9.4. GPIO registers

The GPIO related registers can be written in word, half word and byte mode.

9.4.1. GPIO port mode register (GPIOx_MODER) (x = A, B, F)

Address offset: 0x00

Reset value:

- 0xEBFF FFFF for GPIOA
- 0xFFFF FFFF for GPIOB
- 0xFFFF FCFF for GPIOF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
RW	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
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MODEy[1:0]	RW		y = 15..0 These bits are written by software to configure the I/O mode 00: Input mode 01: General purpose output mode 10: Alternate function mode 11: Analog mode (reset state)

9.4.2. GPIO port output type register (GPIOx_OTYPER) (x = A, B, F)

Address offset: 0x04

Reset value: 0x0000 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	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15:0	MODE[15:0]	RW		These bits are written by software to configure the I/O output type 0: Output push-pull (reset state) 1: Output open-drain

9.4.3. GPIO port output speed register (GPIOx_OSPEEDR) (x = A, B, F)

Address offset: 0x08

Reset value:

- 0x0C00 0000 (for port A)
- 0x0000 0000 (for other ports)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15		OSPEED14		OSPEED13		OSPEED12		OSPEED11		OSPEED10		OSPEED9		OSPEED8	
RW	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
OSPEED7		OSPEED6		OSPEED5		OSPEED4		OSPEED3		OSPEED2		OSPEED1		OSPEED0	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	OSPEEDy[1:0]	RW		y = 15..0 These bits are written by software to configure the I/O output speed 00:Very low speed 01:Low speed 10:High speed 11:Very high speed

9.4.4. GPIO port pull-up and pull-down register (GPIOx_PUPDR) (x = A, B, F)

Address offset: 0x0C

Reset value:

- 0x2400 0000 (for port A)
- 0x0000 0000 (for port B)
- 0x0000 0200 (for port F)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]	
RW	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
PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31 :0	PUPDy[1:0]	RW		y = 15..0 These bits are written by software to configure the I/O pull-up or pull-down 00: No pull-up or pull-down 01: Pull-up 10: Pull-down 11: Reserved

9.4.5. GPIO port input data register (GPIOx_IDR) (x = A, B, F)

Address offset: 0x10

Reset value: 0x0000 XXXX

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
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

Bit	Name	R/W	Reset Value	Function

31:16	Reserved			
15:0	Idy	R		y = 15..0 This is read-only, it contain the input value of the corresponds I/O port

9.4.6. GPIO port output data register (GPIOx_ODR) (x = A, B, F)

Address offset: 0x14

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
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15:0	Ody[1:0]	RW		y = 15..0 These bits are readable and writable by software. Note: For GPIOx_BSRR or GPIOx_BRR registers. (x = A,B,F), each ODR bit can be independently set/cleared.

9.4.7. GPIO port bit set/reset register (GPIOx_BSRR) (x = A, B, F)

Address offset: 0x18

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
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
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit	Name	R/W	Reset Value	Function
31:16	BRy	W		y = 15..0 These bits are write-only. A read to these bits returns the value of 0. 0: No action on the corresponding ODRy bit 1: Clear the corresponding ODRy bit Note: If the corresponding bits of Bsy and Bry are set at the same time, the Bsy bit has priority.
15:0	BSy	W		y = 15..0 These bits are write-only. A read to these bits returns the value of 0. 0: No action on the corresponding ODRy bit 1: Set the corresponding ODRy bit

9.4.8. GPIO port configuration lock register (GPIOx_LCKR) (x = A, B, F)

This register is used to lock the configuration of the port bits when the correct write sequence is applied to bit 16 (LCKK) set. The value of bits [15:0] is used to lock the configuration of the GPIO, the value of LCKR [15:0] must not change. When the LOCK sequence has been applied on the a port bit, the configuration of the port bits cannot be changed until the next system reset.

Note: A special write sequence is used to write the GPIOx_LCKR register. Only word accesses can be performed during the lock sequence.

Each lock bit freezes a specific configuration register (control and alternate function registers)

Address offset: 0x1C

Reset value: 0x0000 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	Res	Res	LCKK
															RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:17	Reserved			
16	LCKK	RW		This bit can be read any time, it can only be modified by the lock key write sequence 0: The port configuration lock key not active 1: The port configuration lock key activated, and the GPIOx_LCKR register is locked until the next system reset LOCK key write sequence: The write sequence of the lock key: write 1- > write 0- > write 1- > read 0- > read 1. The last read can be ignored, but it can be used to confirm that the lock key has been activated. Note: During the LOCK key write sequence, the value of LCK[15:0] must not change. Any error in the lock sequence will stop the lock key from being activated. After the first lock sequence on any bit of the port, any read access on the LCKK will return 1 until the next MCU reset or peripheral reset.
15:0	LCKy	RW		y = 15..0 These bits are readable and writable but can only be written when the LCKK bit is 0. 0: Port configuration not locked 1: Port configuration locked

9.4.9. GPIO alternate function register (low) (GPIOx_AFRL) (x = A, B, F)

Address offset: 0x20

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]			
RW	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
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	AFSELy[3:0] (y = 7 to 0)	RW		These bits are written by software to configure alternate function I/O. AFSELy selection: 0000: AF0 1000: AF8 0001: AF1 1001: AF9 0010: AF2 1010: AF10 0011: AF3 1011: AF11 0100: AF4 1100: AF12 0101: AF5 1101: AF13 0110: AF6 1110: AF14 0111: AF7 1111: AF15

9.4.10. GPIO alternate function register (high) (GPIOx_AFRH) (x = A, B, F)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]			
RW	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**
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	AFSELy[3:0] (y = 8 to 15)	RW		These bits are written by software to configure alternate function I/O. AFSELy selection: 0000: AF0 1000: AF8 0001: AF1 1001: AF9 0010: AF2 1010: AF10 0011: AF3 1011: AF11 0100: AF4 1100: AF12 0101: AF5 1101: AF13 0110: AF6 1110: AF14 0111: AF7 1111: AF15

9.4.11. GPIO port bit reset register (GPIOx_BRR) (x = A, B, F)

Address offset: 0x28

Reset value: 0x0000 0000

31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Res															
15	**14**	**13**	**2**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

Bit	Name	R/W	Reset Value	Function
31:16	Reserved			
15:0	Bry	RW		y = 15..0 These bits are write-only. A read to these bots returns the value of 0. 0: No action on the corresponding Ody bit 1: Clear the corresponding Ody bit

9.4.12. GPIO register map

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**
0x00	GPIOA_MODER	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	1	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	GPIOB_MODER	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	GPIOF_MODER	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	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	1	1	1	1	1	1	1	1

0x04	GPIOF_OTYPER (x = A, B, F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOA_OSPEEDR	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	Reset value	0	0	0	0	1	1	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
0x08	GPIOx_OSPEEDR (x = B, F)	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	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
0x0C	GPIOA_PUPDR	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value	0	0	1	0	0	1	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
0x0C	GPIOB_PUPDR	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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
0x0C	GPIOF_PUPDR	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x10	GPIOx_IDR (x = A, B, F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	Reset value																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x14	GPIOx_ODR (x = A, B, F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (x = A, B, F)	BR15		BR14		BR13		BR12		BR11		BR10		BR9		BR8		BR7		BR6		BR5		BR4		BR3		BR2		BR1		BR0	
	Reset value	0		0		0		0		0		0		0		0		0		0		0		0		0		0		0		0	

0x1C	GPIOx_LCKR (x = A, B, F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFRL (x = A, B, F)	AFSEL7 [3:0]				AFSEL6 [3:0]				AFSEL5 [3:0]				AFSEL4 [3:0]				AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
	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
0x24	GPIOx_AFRH (x = A, B, F)	AFSEL7 [3:0]				AFSEL6 [3:0]				AFSEL5 [3:0]				AFSEL4 [3:0]				AFSEL3 [3:0]				AFSEL2 [3:0]				AFSEL1 [3:0]				AFSEL0 [3:0]			
	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
0x28	GPIOx_BRR (x = A, B, F)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10. System configuration controller (SYSCFG)

The devices feature a set of configuration registers. The main purpose of the system configuration controller are:

- Enable or disable I2C Fast Mode Plus on some IO pins
- Remap some DMA trigger sources to different DMA channels
- Remap the memory located at the beginning of the code area
- Manage the external interrupts connected to GPIOs
- Manage robustness features

10.1. System configuration register

10.1.1. SYSCFG configuration register 1 (SYSCFG_CFGR1)

This register is used for specific configuration of memory and DMA request remap and control special I/O functions.

Two bits are used to configure the type of memory accessible at address 0x0000 0000. These two bits are used to select the physical remap by software, and bypass the hardware BOOT selection. After reset, these bits take the value configured by the actual boot mode.

Address offset: 0x00

Reset value: 0x0000 000x (x is the memory mode selected by the actual boot mode configuration)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	I2C_PF1_ANF	I2C_PF0_ANF	I2C_PB8_ANF	I2C_PB7_ANF	I2C_PB6_ANF	I2C_PA12_ANF	I2C_PA11_ANF	I2C_PA10_ANF	I2C_PA9_ANF	I2C_PA8_ANF	I2C_PA7_ANF	I2C_PA3_ANF	I2C_PA2_ANF	Res	Res
	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**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MEM_MODE[1:0]	
														RW	

Bit	Name	R/W	Reset Value	Function
31	Reserved	RW	-	Read and write
30:18	I2C_IOx_ANF	RW	0	Analog filter enable control of I2C related I/O 0: Analog filter disable 1: Analog filter enable
17:2	Reserved	RW	0	Read and write
1:0	MEM_MODE [1:0]			Memory mapping selection bit Set and clear by software. They control the mapping of memory at address 0x0000 0000. After reset, these bits take on the actual boot mode configuration values. X0: Main Flash, mapped at 0x0000 0000 01: System Flash, mapped at 0x0000 0000 11: SRAM, mapped at 0x0000 0000

10.1.2. SYSCFG configuration register 2 (SYSCFG_CFGR2)

Address offset: 0x18

Reset value: 0x0000

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	ETR_SRC_TIM1		COMP2_BRK_TIM17	COMP1_BRK_TIM17	COMP2_BRK_TIM16	COMP1_BRK_TIM16	COMP2_BRK_TIM1	COMP1_BRK_TIM1	PVD_LOCK	Res	LOCKUP_LOCK
					RW		RW	RW	RW	RW	RW	RW	RW		RW

Bit	Name	R/W	Reset Value	Function
31:11	Reserved	-	-	-
10:9	ETR_SRC_TIM1[1:0]	RW	2'b00	TIMER1 ETR input source selection. 2 'b00: ETR source from GPIO 2 'b01: ETR source from COMP1 2 'b10: ETR source from COMP2 2 'b11: ETR source from ADC
8	COMP2_BRK_TIM17	RW	0	COMP2 as TIMx break input enable. 0: COMP2 output is not used as TIM17 break input 1: COMP2 output as TIM17 break input
7	COMP1_BRK_TIM17	RW	0	COMP1 as TIMx break input enable. 0: COMP1 output is not used as TIM17 break input 1: COMP1 output as TIM17 break input
6	COMP2_BRK_TIM16	RW	0	COMP2 as TIMx break input enable. 0: COMP2 output is not used as TIM16 break input 1: COMP2 output as TIM16 break input
5	COMP1_BRK_TIM16	RW	0	COMP1 as TIMx break input enable. 0: COMP1 output is not used as TIM16 break input 1: COMP1 output as TIM16 break input
4	COMP2_BRK_TIM1	RW	0	COMP2 as TIMx break input enable. 0: COMP2 output is not used as TIM1 break input 1: COMP2 output as TIM1 break input
3	COMP1_BRK_TIM1	RW	0	COMP1 as TIMx break input enable. 0: COMP1 output is not used as TIM1 break input 1: COMP1 output as TIM1 break input
2	PVD_LOCK	RW	0	PVD Lock enable bit Set by software and cleared by system reset. It can be used as the brake input to enable and lock the PVD connection to TIM1/TIM16/TIM17, and also lock the PVDE of the PWR_CR register. 0: PVD interrupt is not connected to the brake input of TIM1/TIM16/TIM17. The PVDE bit can be written by the application. 1: PVD interrupt is connected to the brake input of TIM1/TIM16/TIM17. The PVDE bit is read only.
1	Reserved	-	-	-
0	LOCKUP_LOCK	RW		Cortex-M0+ LOCKUP enable bit Set by software and cleared by system reset. It can enable and lock the LOCKUP (hardfault) output of Cortex-M0+ to the brake input of TIM1/TIM16/TIM17. 0: The LOCKUP output of Cortex-M0+ is not connected to the brake input of TIM1/TIM16/TIM17 1: The LOCKUP output of Cortex-M0+ is connected to the brake input of TIM1/TIM16/TIM17

10.1.3. SYSCFG configuration register 3 (SYSCFG_CFGR3)

Address offset: 0x1C

Reset value: 0x0000 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	DMA3_ ACK-LVL	DMA3_MAP				
										RW	RW	RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	DMA2_ ACK-LVL	DMA2_MAP					Res	Res	DMA1_ ACK-LVL	DMA1_MAP				
		RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:22	Reserved			
21	DMA3_ACKLVL	RW	0	Response speed enable bit for DMA channel 3 0: Normal speed response 1: Quick speed response
20:16	DMA3_MAP	RW	0	00000: ADC 00001: SPI1_TX 00010: SPI1_RX 00011: SPI2_TX 00100: SPI2_RX 00101: USART1_TX 00110: USART 1_RX 00111: USART2_TX 01000: USART2_RX 01001: I2C_TX 01010: I2C_RX 01011: TIM1_CH1 01100: TIM1_CH2 01101: TIM1_CH3 01110: TIM1_CH4 01111: TIM1_COM, 10000: TIM1_UP 10001: TIM1_TRIG 10010: TIM3_CH1 10011: TIM3_CH3 10100: TIM3_CH4 10101: TIM3_TRG 10110: TIM3_UP 10111: Reserved 11000: TIM16_CH1 11001: TIM16_UP 11010: TIM1 7_CH1 11011: TIM17_UP Others: Reserved
15:14	Reserved			
13	DMA2_ACKLVL	RW	0	Response speed enable bit for DMA channel 2 0: Normal speed response 1: Quick speed response
12:8	DMA2_MAP	RW	0	00000: ADC 00001: SPI1_TX 00010: SPI1_RX 00011: SPI2_TX 00100: SPI2_RX 00101: USART1_TX 00110: USART 1_RX 00111: USART2_TX 01000: USART2_RX 01001: I2C_TX 01010: I2C_RX 01011: TIM1_CH1 01100: TIM1_CH2 01101: TIM1_CH3 01110: TIM1_CH4 01111: TIM1_COM 10000: TIM1_UP 10001: TIM1_TRIG

				10010: TIM3_CH1 10011: TIM3_CH3 10100: TIM3_CH4 10101: TIM3_TRG 10110: TIM3_UP 10111: Reserved 11000: TIM16_CH1 11001: TIM16_UP 11010: TIM17_CH1 11011: TIM17_UP Others: Reserved
7:6	Reserved			
5	DMA1_ACKLVL	RW	0	Response speed enable bit for DMA channel 1 0: Normal speed response 1: Quick speed response
4:0	DMA1_MAP	RW	0	00000: ADC 00001: SPI1_TX 00010: SPI1_RX 00011: SPI2_TX 00100: SPI2_RX 00101: USART1_TX 00110: USART 1_RX 00111: USART2_TX 01000: USART2_RX 01001: I2C_TX 01010: I2C_RX 01011: TIM1_CH1 01100: TIM1_CH2 01101: TIM1_CH3 01110: TIM1_CH4 01111: TIM1_COM 10000: TIM1_UP 10001: TIM1_TRIG 10010: TIM3_CH1 10011: TIM3_CH3 10100: TIM3_CH4 10101: TIM3_TRG 10110: TIM3_UP 10111: Reserved 11000: TIM16_CH1 11001: TIM16_UP 11010: TIM17_CH1 11011: TIM17_UP Others: Reserved

10.1.4. SYSCFG register map

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
0x00	SYSCFG_CFGR1	Res.	I2C_PF1_ANF	I2C_PF0_ANF	I2C_PB8_ANF	I2C_PB7_ANF	I2C_PB6_ANF	I2C_PA12_ANF	I2C_PA11_ANF	I2C_PA10_ANF	I2C_PA9_ANF	I2C_PA8_ANF	I2C_PA7_ANF	I2C_PA3_ANF	I2C_PA2_ANF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM_MODE[1:0]	
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0																	X	X

		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
0x18	SYSCFG_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETR_SRC_TIM1[1:0]		COMP2_BRK_TIM17	COMP1_BRK_TIM17	COMP2_BRK_TIM16	COMP1_BRK_TIM16	COMP2_BRK_TIM21	COMP2_BRK_TIM22	COMP2_BRK_TIM23	Res.	LOCKUP_LOCK
	Reset value																							0	0	0	0	0	0	0	0		0
0x1C	SYSCFG_CFGR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMA3_ACKLVL	DMA3_MAP[4:0]						Res.	Res.	DMA2_ACKLVL	DMA2_MAP[4:0]						Res.	Res.	DMA1_ACKLVL	DMA1_MAP[4:0]					
	Reset value												0	0	0	0	0				0	0	0	0	0	0				0	0	0	0	0	0

11. DMA

11.1. DMA introduction

Direct memory access (DMA) is used in order to provide high-speed data transfer between peripherals and memory as well as memory to memory. Data can be quickly moved by DMA without any CPU actions. This keeps CPU resources free for other operations.

The two DMA controllers have 3 channels in total, each dedicated to managing memory access requests from one or more peripherals. It has an arbiter for handling the priority between DMA requests.

11.2. DMA main features

- Three independently configurable channels (requests)
- Each of the 12 channels is connected to dedicated hardware DMA requests, software trigger is also supported on each channel. This configuration is done by software
- Priorities between requests from channels of one DMA are software programmable (4 levels consisting of very high, high, medium, low) or hardware in case of equality (request 1 has priority over request 2, etc.)
- Independent source and destination transfer size (byte, half word, word), emulating packing and unpacking. Source/destination addresses must be aligned on the data size
- Three event flags (DMA Half Transfer, DMA Transfer complete and DMA Transfer Error) logically ORed together in a single interrupt request for each channel
- Memory-to-memory transfer
- Peripheral-to-memory and memory-to-peripheral, and peripheral-to-peripheral transfers
- Access to Flash, SRAM, APB and AHB peripherals as source and destination
- Programmable number of data to be transferred: up to 65536

11.3. DMA functional description

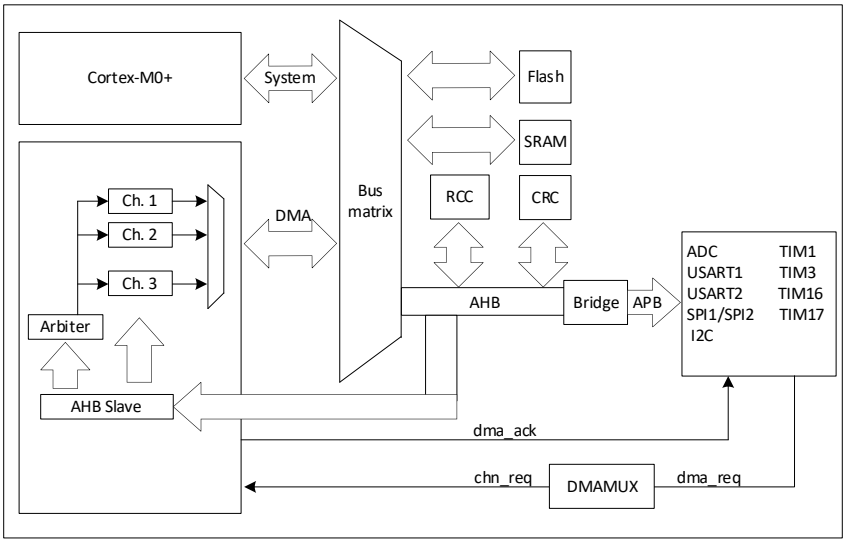


Figure 11-1 DMA block diagram

11.3.1. DMA transactions

After an event, the peripheral sends a request signal to the DMA Controller. The DMA controller serves the request depending on the channel priorities. As soon as the DMA Controller accesses the peripheral, an Acknowledge is sent to the peripheral by the DMA Controller. The peripheral releases its request as soon as it gets the Acknowledge from the DMA Controller. Once the request is deasserted by the peripheral, the DMA Controller release the Acknowledge. If there are more requests, the peripheral can initiate the next transaction. In summary, each DMA transfer consists of three operations:

- The loading of data from the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the DMA_CPARx or DMA_CMARx register
- The storage of the data loaded to the peripheral data register or a location in memory addressed through an internal current peripheral/memory address register. The start address used for the first transfer is the base peripheral/memory address programmed in the DMA_CPARx or DMA_CMARx register.
- The post-decrementing of the DMA_CNDTRx register, which contains the number of transactions that have still to be performed.

11.3.2. Arbiter

The arbiter manages the channel requests based on their priority and launches the peripheral/memory access sequences.

The priorities are managed in two stages:

- Software: each channel priority can be configured in the DMA_CCRx register. There are four levels:
  - Very high priority
  - High priority
  - Medium priority
  - Low priority
- Hardware: if 2 requests have the same software priority level, the channel with the lowest number will get priority versus the channel with the highest number. For example, channel 2 gets priority over channel 4.

11.3.3. DMA channels

Each channel can handle DMA transfer between a peripheral register located at a fixed address and a memory address. The amount of data to be transferred (up to 65535) is programmable. The register which contains the amount of data items to be transferred is decremented after each transaction.

Programmable data sizes

Transfer data sizes of the peripheral and memory are fully programmable through the PSIZE and MSIZE bits in the DMA_CCRx register.

Pointer incrementation

Peripheral and memory pointers can optionally be automatically post-incremented after each transaction depending on the PINC and MINC bits in the DMA_CCRx register. If incremented mode is enabled, the address of the next transfer will be the address of the previous one incremented by 1, 2 or 4 depending on the chosen data size.

The first transfer address is the one programmed in the DMA_CPARx/DMA_CMARx registers. During transfer operations, these registers keep the initially programmed value. The current transfer addresses (in the current internal peripheral/memory address register) are not accessible by software.

If the channel is configured in noncircular mode, no DMA request is served after the last transfer (that is once the number of data items to be transferred has reached zero). In order to reload a new number of data items to be transferred into the DMA_CNDTRx register, the DMA channel must be disabled.

In circular mode, after the last transfer, the DMA_CNDTRx register is automatically reloaded with the initially programmed value. The current internal address registers are reloaded with the base address values from the DMA_CPARx/DMA_CMARx registers.

Circular mode

Circular mode is available to handle circular buffers and continuous data flows (e.g. ADC scan mode). This feature can be enabled using the CIRC bit in the DMA_CCRx register.When circular mode is activated, the number of data to be transferred is automatically reloaded with the initial value programmed during the channel configuration phase, and the DMA requests continue to be served.

Memory-to-memory mode

The DMA channels can also work without being triggered by a request from a peripheral.This mode is called Memory to Memory mode.

If the MEM2MEM bit in the DMA_CCRx register is set, then the channel initiates transfers as soon as it is enabled by software by setting the Enable bit (EN) in the DMA_CCRx register. The transfer stops once the DMA_CNDTRx register reaches zero. Memory to Memory mode may not be used at the same time as Circular mode.

Channel configuration procedure

The following sequence should be followed to configure a DMA channelx (where x is the channel number).

- Set the peripheral register address in the DMA_CPARx register. The data will be moved from/ to this address to/ from the memory after the peripheral event.
- Set the memory address in the DMA_CMARx register. The data will be written to or read from this memory after the peripheral event.
- Configure the total number of data to be transferred in the DMA_CNDTRx register. After each peripheral event, this value will be decremented.
- Configure the channel priority using the PL[1:0] bits in the DMA_CCRx register.
- Configure data transfer direction, circular mode, peripheral & memory incremented mode, peripheral & memory data size, and interrupt after half and/or full transfer in the DMA_CCRx register.
- Activate the channel by setting the ENABLE bit in the DMA_CCRx register.

As soon as the channel is enabled, it can serve any DMA request from the peripheral connected on the channel. Once half of the bytes are transferred, the half-transfer flag (HTIF) is set and an interrupt is generated if the Half-Transfer Interrupt Enable bit (HTIE) is set. At the end of the transfer, the Transfer Complete Flag (TCIF) is set and an interrupt is generated if the Transfer Complete Interrupt Enable bit (TCIE) is set.

11.3.4. Programmable data width, data alignment and endians

When PSIZE and MSIZE are not equal, the DMA performs some data alignments as described in Table 11-1.

Table 11-1 Programmable data width and endian behavior (when bits PINC = MINC = 1)

Source port width	Des-tina-tion port width	Number of data items to transfer (NDT)	Source con-tent: address / data	Transfer operations	Destination content: address / data
8	8	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1: READ B0[7:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B1[7:0] @0x1 then WRITE B1[7:0] @0x1 3: READ B2[7:0] @0x2 then WRITE B2[7:0] @0x2 4: READ B3[7:0] @0x3 then WRITE B3[7:0] @0x3	0x0/B0 0x1/B1 0x2/B2 0x3/B3
8	16	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1: READ B0[7:0] @0x0 then WRITE 00B0[15:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 00B1[15:0] @0x2 3: READ B3[7:0] @0x2 then WRITE 00B2[15:0] @0x4 4: READ B4[7:0] @0x3 then WRITE 00B3[15:0] @0x6	0x0/00B0 0x2/00B1 0x4/00B2 0x6/00B3
8	32	4	0x0/B0 0x1/B1 0x2/B2 0x3/B3	1: READ B0[7:0] @0x0 then WRITE 000000B0[31:0] @0x0 2: READ B1[7:0] @0x1 then WRITE 000000B1[31:0] @0x4 3: READ B3[7:0] @0x2 then WRITE 000000B2[31:0] @0x8 4: READ B4[7:0] @0x3 then WRITE 000000B3[31:0] @0xC	0x0/000000B0 0x4/000000B1 0x8/000000B2 0xC/000000B3
16	8	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1: READ B1B0[15:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B2[7:0] @0x1 3: READ B5B4[15:0] @0x4 then WRITE B4[7:0] @0x2 4: READ B7B6[15:0] @0x6 then WRITE B6[7:0] @0x3	0x0/B0 0x1/B2 0x2/B4 0x3/B6
16	16	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1: READ B1B0[15:0] @0x0 then WRITE B1B0[15:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE B3B2[15:0] @0x2 3: READ B5B4[15:0] @0x4 then WRITE B5B4[15:0] @0x4 4: READ B7B6[15:0] @0x6 then WRITE B7B6[15:0] @0x6	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6
16	32	4	0x0/B1B0 0x2/B3B2 0x4/B5B4 0x6/B7B6	1: READ B1B0[15:0] @0x0 then WRITE 0000B1B0[31:0] @0x0 2: READ B3B2[15:0] @0x2 then WRITE 0000B3B2[31:0] @0x4 3: READ B5B4[15:0] @0x4 then WRITE 0000B5B4[31:0] @0x8 4: READ B7B6[15:0] @0x6 then WRITE 0000B7B6[31:0] @0xC	0x0/0000B1B0 0x4/000B3B2 0x8/0000B5B4 0xC/0000B7B6
32	8	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEB-DBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BC[7:0] @0x3	0x0/B0 0x1/B4 0x2/B8 0x3/BC
32	16	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8 0xC/BFBEB-DBC	1: READ B3B2B1B0[31:0] @0x0 then WRITE B1B0[7:0] @0x0 2: READ B7B6B5B4[31:0] @0x4 then WRITE B5B4[7:0] @0x1 3: READ BBBAB9B8[31:0] @0x8 then WRITE B9B8[7:0] @0x2 4: READ BFBEBDBC[31:0] @0xC then WRITE BDBC[7:0] @0x3	0x0/B1B0 0x2/B5B4 0x4/B9B8 0x6/BDBC
32	32	4	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8	1: READ B3B2B1B0[31:0] @0x0 then WRITE B3B2B1B0[31:0] @0x0	0x0/B3B2B1B0 0x4/B7B6B5B4 0x8/BBBAB9B8

			0xC/BFBEB-DBC	2: READ B7B6B5B4[31:0] @0x4 then WRITE B7B6B5B4[31:0] @0x4 3: READ BBBAB9B8[31:0] @0x8 then WRITE BBBAB9B8[31:0] @0x8 4: READ BFBEBDBC[31:0] @0xC then WRITE BFBEBDBC[31:0] @0xC	0xC/BFBEBDBC

Addressing an AHB peripheral that does not support byte or halfword write operations

When the DMA initiates an AHB byte or halfword write operation, the data are duplicated on the unused lanes of the HWDATA[31:0] bus. So when the used AHB slave peripheral does not support byte or halfword write operations (when HSIZE is not used by the peripheral) and does not generate any error, the DMA writes the 32 HWDATA bits as shown in the two examples below:

- To write the halfword “0xABCD”, the DMA sets the HWDATA bus to “0xABCDABCD” with HSIZE = Half-Word
- To write the byte “0xAB”, the DMA sets the HWDATA bus to “0xABABABAB” with HSIZE = Byte

Assuming that the AHB/APB bridge is an AHB 32-bit slave peripheral that does not take the HSIZE data into account, it will transform any AHB byte or halfword operation into a 32-bit APB operation in the following manner:

- An AHB byte write operation of the data “0xB0” to 0x0 (or to 0x1, 0x2 or 0x3) will be converted to an APB word write operation of the data “0xB0B0B0B0” to 0x0
- An AHB halfword write operation of the data “0xB1B0” to 0x0 (or to 0x2) will be converted to an APB word write operation of the data “0xB1B0B1B0” to 0x0

For instance, to write the APB backup registers (16-bit registers aligned to a 32-bit address boundary), the memory source size (MSIZE) must be configured to “16-bit” and the peripheral destination size (PSIZE) to “32-bit”.

11.3.5. Error management

A DMA transfer error can be generated by reading from or writing to a reserved address space. When a DMA transfer error occurs during a DMA read or a write access, the faulty channel is automatically disabled through a hardware clear of its EN bit in the corresponding Channel configuration register (DMA_CCRx). The channel's transfer error interrupt flag (TEIF) in the DMA_IFR register is set and an interrupt is generated if the transfer error interrupt enable bit (TEIE) in the DMA_CCRx register is set.

11.3.6. DMA Interrupts

An interrupt can be produced on a Half-transfer, Transfer complete or Transfer error for each DMA channel. Separate interrupt enable bits are available for flexibility.

Table 11-2 DMA interrupt requests

Interrput event	Event flag	Enable Control bit
Half-transfer	HTIF	HTIE
Transfer complete	TCIF	TCIE
Transfer error	TEIF	TEIE

Note: When the DMA_CNDTRx register is 1, the HTIFx bit will not be set, and the TCIFx bit will be set when the transfer is complete.

11.3.7. DMA request mapping

DMA controller

Table 11-3 Summary of DMA1 requests for each channel

Peripherals	Channel 1	Channel 2	Channel 3
ADC	ADC	ADC	ADC
SPI	SPI_RX SPI_TX	SPI_RX SPI_TX	SPI_RX SPI_TX
USART	USART1_RX USART1_TX USART2_RX USART2_TX	USART1_RX USART1_TX USART2_RX USART2_TX	USART1_RX USART1_TX USART2_RX USART2_TX
I2C	I2C_RX I2C_TX	I2C_RX I2C_TX	I2C_RX I2C_TX
TIM1	TIM1_CH1 TIM1_CH2 TIM1_CH3 TIM1_CH4 TIM1_UP TIM1_TRIG TIM1_COM	TIM1_CH1 TIM1_CH2 TIM1_CH3 TIM1_CH4 TIM1_UP TIM1_TRIG TIM1_COM	TIM1_CH1 TIM1_CH2 TIM1_CH3 TIM1_CH4 TIM1_UP TIM1_TRIG TIM1_COM
TIM3	TIM3_CH1 TIM3_CH2 TIM3_CH3 TIM3_CH4 TIM3_UP TIM3_TRIG	TIM3_CH1 TIM3_CH2 TIM3_CH3 TIM3_CH4 TIM3_UP TIM3_TRIG	TIM3_CH1 TIM3_CH2 TIM3_CH3 TIM3_CH4 TIM3_UP TIM3_TRIG
TIM16	TIM16_CH1 TIM16_UP	TIM16_CH1 TIM16_UP	TIM16_CH1 TIM16_UP
TIM17	TIM17_CH1 TIM17_UP	TIM17_CH1 TIM17_UP	TIM17_CH1 TIM17_UP

11.4. DMA registers

11.4.1. DMA interrupt status register (DMA_ISR)

Address offset: 0x00

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
				R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11	TEIF3	R	0	Channel 3 transfer error flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer error (TE) on channel 3 (TE) 1: A transfer error (TE) occurred on channel 3 (TE)
10	HTIF3	R	0	Channel 3 half transfer flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No half transfer (HT) event on channel 3 1: A half transfer (HT) event occurred on channel 3
9	TCIF3	R	0	Channel 3 transfer complete flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer complete (TC) event on channel 3

				1: A transfer complete (TC) event occurred on channel 3
8	GIF3	R	0	Channel 3 global interrupt flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No TE, HT or TC event on channel 3 1: A TE, HT or TC event occurred on channel 3
7	TEIF2	R	0	Channel 2 transfer error flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer error (TE) on channel 2 (TE) 1: A transfer error (TE) occurred on channel 2 (TE)
6	HTIF2	R	0	Channel 2 half transfer flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No half transfer (HT) event on channel 2 1: A half transfer (HT) event occurred on channel 2
5	TCIF2	R	0	Channel 2 transfer complete flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer complete (TC) event on channel 2 1: A transfer complete (TC) event occurred on channel 2
4	GIF2	R	0	Channel 2 global interrupt flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No TE, HT or TC event on channel 2 1: A TE, HT or TC event occurred on channel 2
3	TEIF1	R	0	Channel 1 transfer error flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer error (TE) on channel 1 (TE) 1: A transfer error (TE) occurred on channel 1 (TE)
2	HTIF1	R	0	Channel 1 half transfer flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No half transfer (HT) event on channel 1 1: A half transfer (HT) event occurred on channel 1
1	TCIF1	R	0	Channel 1 transfer complete flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No transfer complete (TC) event on channel 1 1: A transfer complete (TC) event occurred on channel 1
0	GIF1	R	0	Channel 1 global interrupt flag This bit is set by hardware. It is cleared by software writing 1 to the corresponding bit in the DMA_IFCR register. 0: No TE, HT or TC event on channel 1 1: A TE, HT or TC event occurred on channel 1

11.4.2. DMA interrupt flag clear register (DMA_IFCR)

Address offset: 0x04

Reset value: 0x0000 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	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Res	Res	Res	Res	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
				W	W	W	W	W	W	W	W	W	W	W	W

Bit	Name	R/W	Reset Value	Function
31:12	Reserved	-	-	Reserved
11	CTEIF3	W	0	Channel 3 transfer error clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TEIF flag in the DMA_ISR register
10	CHTIF3	W	0	Channel 3 half transfer clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding HTIF flag in the DMA_ISR register
9	CTCIF3	W	0	Channel 3 transfer complete clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TCIF flag in the DMA_ISR register
8	CGIF3	W	0	Channel 3 global interrupt clear This bit is set and cleared by software. 0: No effect 1: Clears the GIF, TEIF, HTIF and TCIF flags in the DMA_ISR register
7	CTEIF2	W	0	Channel 2 transfer error clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TEIF flag in the DMA_ISR register
6	CHTIF2	W	0	Channel 2 half transfer clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding HTIF flag in the DMA_ISR register
5	CTCIF2	W	0	Channel 2 transfer complete clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TCIF flag in the DMA_ISR register
4	CGIF2	W	0	Channel 2 global interrupt clear This bit is set and cleared by software. 0: No effect 1: Clears the GIF, TEIF, HTIF and TCIF flags in the DMA_ISR register
3	CTEIF1	W	0	Channel 1 transfer error clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TEIF flag in the DMA_ISR register
2	CHTIF1	W	0	Channel 1 half transfer clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding HTIF flag in the DMA_ISR register
1	CTCIF1	W	0	Channel 1 transfer complete clear This bit is set and cleared by software. 0: No effect 1: Clears the corresponding TCIF flag in the DMA_ISR register
0	CGIF1	W	0	Channel 1 global interrupt clear This bit is set and cleared by software. 0: No effect 1: Clears the GIF, TEIF, HTIF and TCIF flags in the DMA_ISR register

11.4.3. DMA channel 1 configuration register (DMA_CCR1)

Address offset: 0x08

Reset value: 0x0000 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	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 1 Memory to memory mode This bit is set and cleared by software. 0: Memory to memory mode disabled 1: Memory to memory mode enabled
13:12	PL[1:0]	RW	0	Channel 1 priority level These bits are set and cleared by software. 00: Low 01: Medium 10: High 11: Very high
11:10	MSIZE[1:0]	RW	0	Channel 1 Memory size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
9:8	PSIZE[1:0]	RW	0	Channel 1 Peripheral size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
7	MINC	RW	0	Channel 1 Memory increment mode This bit is set and cleared by software. 0: Memory increment mode disabled 1: Memory increment mode enabled
6	PINC	RW	0	Channel 1 Peripheral increment mode This bit is set and cleared by software. 0: Peripheral increment mode disabled 1: Peripheral increment mode enabled
5	CIRC	RW	0	Channel 1 Circular mode This bit is set and cleared by software. 0: Circular mode disabled 1: Circular mode enabled
4	DIR	RW	0	Channel 1 Data transfer direction This bit is set and cleared by software. 0: Read from peripheral 1: Read from memory
3	TEIE	RW	0	Channel 1 Transfer error interrupt enable This bit is set and cleared by software. 0: TE interrupt disabled 1: TE interrupt enabled
2	HTIE	RW	0	Channel 1 Half transfer interrupt enable This bit is set and cleared by software. 0: HT interrupt disabled 1: HT interrupt enabled
1	TCIE	RW	0	Channel 1 Transfer complete interrupt enable This bit is set and cleared by software. 0: TC interrupt disabled 1: TC interrupt enabled
0	EN	RW	0	Channel 1 Channel enable This bit is set and cleared by software. 0: Channel disabled

				1: Channel enabled

11.4.4. DMA channel 1 number of data register (DMA_CNDTR1)

Address offset: 0x0C

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT[15:0]	RW	0	channel 1 Number of data to transfer Number of data to be transferred (0 up to 65535). This register can only be written when the channel is disabled. Once the channel is enabled, this register is read-only, indicating the remaining bytes to be transmitted. This register decrements after each DMA transfer. Once the transfer is completed, this register can either stay at zero or be reloaded automatically by the value previously programmed if the channel is configured in autoreload mode. If this register is zero, no transaction can be served whether the channel is enabled or not.

11.4.5. DMA channel 1 peripheral address register (DMA_CPAR1)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA[31:16]															
RW	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**
PA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA[31:0]	RW	0	channel 1 Peripheral address Base address of the peripheral data register from/to which the data will be read/written. When PSIZE is 01 (16-bit), the PA[0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE is 10 (32-bit), PA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.6. DMA channel 1 memory address register (DMA_CMAR1)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA[31:16]															
RW	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**
MA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA[31:0]	RW	0	Channel 1 Memory address Base address of the memory area from/to which the data will be read/written. When MSIZE is 01 (16-bit), the MA[0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE is 10 (32-bit), MA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.7. DMA channel 2 configuration register (DMA_CCR2)

Address offset: 0x1C

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 2 set and cleared by software. 0: Memory to memory mode disabled 1: Memory to memory mode enabled
1312	PL[1:0]	RW	0	Channel 2 priority level These bits are set and cleared by software. 00: Low 01: Medium 10: High 11: Very high
11:10	MSIZE[1:0]	RW	0	Channel 2 Memory size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
9:8	PSIZE[1:0]	RW	0	Channel 2 Peripheral size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
7	MINC	RW	0	Channel 2 Memory increment mode This bit is set and cleared by software. 0: Memory increment mode disabled 1: Memory increment mode enabled
6	PINC	RW	0	Channel 2 Peripheral increment mode This bit is set and cleared by software. 0: Peripheral increment mode disabled 1: Peripheral increment mode enabled
5	CIRC	RW	0	Channel 2 Circular mode This bit is set and cleared by software. 0: Circular mode disabled 1: Circular mode enabled
4	DIR	RW	0	Channel 2 Data transfer direction This bit is set and cleared by software. 0: Read from peripheral 1: Read from memory
3	TEIE	RW	0	Channel 2 Transfer error interrupt enable This bit is set and cleared by software. 0: TE interrupt disabled 1: TE interrupt enabled

2	HTIE	RW	0	Channel 2 Half transfer interrupt enable This bit is set and cleared by software. 0: HT interrupt disabled 1: HT interrupt enabled
1	TCIE	RW	0	Channel 2 Transfer complete interrupt enable This bit is set and cleared by software. 0: TC interrupt disabled 1: TC interrupt enabled
0	EN	RW	0	Channel 2 Channel enable This bit is set and cleared by software. 0: Channel disabled 1: Channel enabled

11.4.8. DMA channel 2 number of data register (DMA_CNDTR2)

Address offset: 0x20

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:16	Reserved	-	-	Reserved
15:0	NDT[15:0]	RW	0	channel 2 Number of data to transfer Number of data to be transferred (0 up to 65535). This register can only be written when the channel is disabled. Once the channel is enabled, this register is read-only, indicating the remaining bytes to be transmitted. This register decrements after each DMA transfer. Once the transfer is completed, this register can either stay at zero or be reloaded automatically by the value previously programmed if the channel is configured in autoreload mode. If this register is zero, no transaction can be served whether the channel is enabled or not.

11.4.9. DMA channel 2 peripheral address register (DMA_CPAR2)

Address offset: 0x24

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA[31:16]															
RW	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**
PA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA[31:0]	RW	0	channel 2 Peripheral address Base address of the peripheral data register from/to which the data will be read/written. When PSIZE is 01 (16-bit), the PA[0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE is 10 (32-bit), PA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.10. DMA channel 2 memory address register (DMA_CMAR2)

Address offset: 0x28

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA[31:16]															
RW	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**
MA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA[31:0]	RW	0	Channel 2 Memory address Base address of the memory area from/to which the data will be read/written. When MSIZE is 01 (16-bit), the MA[0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE is 10 (32-bit), MA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.11. DMA channel 3 configuration register (DMA_CCR3)

Address offset: 0x30

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:15	Reserved	-	-	Reserved
14	MEM2MEM	RW	0	Channel 3 set and cleared by software. 0: Memory to memory mode disabled 1: Memory to memory mode enabled
13:12	PL[1:0]	RW	0	Channel 3 priority level These bits are set and cleared by software. 00: Low 01: Medium 10: High 11: Very high
11:10	MSIZE[1:0]	RW	0	Channel 3 Memory size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
9:8	PSIZE[1:0]	RW	0	Channel 3 Peripheral size These bits are set and cleared by software. 00: 8-bits 01: 16-bits 10: 32-bits 11: Reserved
7	MINC	RW	0	Channel 3 Memory increment mode This bit is set and cleared by software. 0: Memory increment mode disabled 1: Memory increment mode enabled
6	PINC	RW	0	Channel 3 Peripheral increment mode This bit is set and cleared by software. 0: Peripheral increment mode disabled 1: Peripheral increment mode enabled
5	CIRC	RW	0	Channel 3 Circular mode

				This bit is set and cleared by software. 0: Circular mode disabled 1: Circular mode enabled
4	DIR	RW	0	Channel 3 Data transfer direction This bit is set and cleared by software. 0: Read from peripheral 1: Read from memory
3	TEIE	RW	0	Channel 3 Transfer error interrupt enable This bit is set and cleared by software. 0: TE interrupt disabled 1: TE interrupt enabled
2	HTIE	RW	0	Channel 3 Half transfer interrupt enable This bit is set and cleared by software. 0: HT interrupt disabled 1: HT interrupt enabled
1	TCIE	RW	0	Channel 3 Transfer complete interrupt enable This bit is set and cleared by software. 0: TC interrupt disabled 1: TC interrupt enabled
0	EN	RW	0	Channel 3 Channel enable This bit is set and cleared by software. 0: Channel disabled 1: Channel enabled

11.4.12. DMA channel 3 number of data register (DMA_CNDTR3)

Address offset: 0x34

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
NDT[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31: 16	Reserved	-	-	Reserved
15: 0	NDT[15:0]	RW	0	channel 3 Number of data to transfer Number of data to be transferred (0 up to 65535). This register can only be written when the channel is disabled. Once the channel is enabled, this register is read-only, indicating the remaining bytes to be transmitted. This register decrements after each DMA transfer. Once the transfer is completed, this register can either stay at zero or be reloaded automatically by the value previously programmed if the channel is configured in autoreload mode. If this register is zero, no transaction can be served whether the channel is enabled or not.

11.4.13. DMA channel 3 peripheral address register (DMA_CPAR3)

Address offset: 0x38

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PA[31:16]															
RW	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**
PA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	PA[31:0]	RW	0	channel 3 Peripheral address

				Base address of the peripheral data register from/to which the data will be read/written. When PSIZE is 01 (16-bit), the PA[0] bit is ignored. Access is automatically aligned to a half-word address. When PSIZE is 10 (32-bit), PA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.14. DMA channel 3 memory address register (DMA_CMAR3)

Address offset: 0x3C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MA[31:16]															
RW	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**
MA[15:0]															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:0	MA[31:0]	RW	0	Channel 3 Memory address Base address of the memory area from/to which the data will be read/written. When MSIZE is 01 (16-bit), the MA[0] bit is ignored. Access is automatically aligned to a half-word address. When MSIZE is 10 (32-bit), MA[1:0] are ignored. Access is automatically aligned to a word address.

11.4.15. DMA register map

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
0x00	DMA_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x04	DMA_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEIF3	CHTIF	CTCIF	CGIF3	CTEIF2	CHTIF	CTCIF	CGIF2	CTEIF1	CHTIF	CTCIF	CGIF1
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x08	DMA_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[9:8]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	DMA_CNDTR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	DMA_CPAR1	PA[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
0x14	DMA_CMAR1	MA[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
0x18	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.
0x1C	DMA_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[9:8]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DMA_CNDTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]																
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	DMA_CPAR2	PA[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
0x28	DMA_CMAR2	MA[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
0x2C	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.
0x30	DMA_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[9:8]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	DMA_CNDTR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]																
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	DMA_CPAR3	PA[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

0x3C	DMA_CMAR3	MA[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

12. Interrupts and events

12.1. Nested vectored interrupt controller (NVIC)

12.1.1. NVIC main features

- 32 maskable interrupt channels (not including the 16 ARM® Cortex®-M0 interrupt lines)
- 4 programmable priority levels (2 bits of interrupt priority are used)
- Low-latency exception and interrupt handling
- Power management control
- Implementation of System Control Registers

The NVIC and the processor core interface are closely coupled, which enables low latency interrupt processing and efficient processing of late arriving interrupts. All interrupts including the core exceptions are managed by the NVIC.

12.1.2. SysTick calibration value register

The SysTick calibration value is set to 6000, which gives a reference time base of 1 ms with the SysTick clock set to 6 MHz (max fHCLK/8).

12.1.3. Interrupt and exception vectors

Position	Priority	Type of priority	Acronym	Description	Address
-	-	-	-	Reserved	0x0000_0000
-	-3	fixed	Reset	Reset	0x0000_0004
-	-2	fixed	NMI_Handler	Non maskable interrupt. The RCC Clock Security System (CSS) is linked to the NMI vector.	0x0000_0008
-	-1	fixed	HardFualt_Handler	All class of fault	0x0000_000C
-	3	settable	SVCall	System service call via SWI instruction	0x0000_002C
-	5	settable	PendSV	Pendable request for system service	0x0000_0038
-	6	settable	SysTick	System tick timer	0x0000_003C
0	7	settable	WWDG	Window watchdog interrupt	0x0000_0040
1	8	settable	PVD	Supply voltage detection interrupt (EXTI line 16)	0x0000_0044
2	9	settable	RTC	RTC interrupt (combined EXTI lines 19)	0x0000_0048
3	10	settable	Flash	Flash global interrupt	0x0000_004C
4	11	settable	RCC	RCC global interrupt	0x0000_0050
5	12	settable	EXTI0_1	EXTI line[1:0] interrupt	0x0000_0054
6	13	settable	EXTI2_3	EXTI line[3:2] interrupt	0x0000_0058
7	14	settable	EXTI4_15	EXTI line[15:4] interrupt	0x0000_005C
8	15	-	Reserved	Reserved	0x0000_0060
9	16	settable	DMA_Channel1	DMA channel 1 interrupt	0x0000_0064
10	17	settable	DMA_Channel2_3	DMA channel 2& channel 3 interrupt	0x0000_0068
11	18	-	Reserved	Reserved	0x0000_006C
12	19	settable	ADC_COMP	ADC and COMP interrupts (COMP combined with EXTI 17 & 18)	0x0000_0070
13	20	settable	TIM1_BRK_UP_TRG_COM	TIM1 break, update, trigger and commutation interrupt	0x0000_0074
14	21	settable	TIM1_CC	TIM1 capture compare interrupt	0x0000_0078
15	22	-	Reserved	Reserved	0x0000_007C
16	23	settable	TIM3	TIM3 global interrupt	0x0000_0080
17	24	settable	LPTIM1	LPTIM interrupt	0x0000_0084
18	25	-	Reserved	Reserved	0x0000_0088

19	26	settable	TIM14	TIM14 global interrupt	0x0000_008C
20	27	-	Reserved	Reserved	0x0000_0090
21	28	settable	TIM16	TIM16 global interrupt	0x0000_0094
22	29	settable	TIM17	TIM17 global interrupt	0x0000_0098
23	30	settable	I2C1	I2C1 global interrupt	0x0000_009C
24	31	-	Reserved	Reserved	0x0000_00A0
25	32	settable	SPI1	SPI1 global interrupt	0x0000_00A4
26	33	settable	SPI2	SPI2 global interrupt	0x0000_00A8
27	34	settable	USART1	USART1 global interrupt	0x0000_00AC
28	35	settable	USART2	USART2 global interrupt	0x0000_00B0
29	36	-	-	-	0x0000_00B4
30	37	settable	LED	LED interrupt	0x0000_00B8
31	38	-	Reserved	Reserved	0x0000_00BC

- The grayed cells (the address less than 0x0000 0040) correspond to the Cortex®-M0+ interrupts.

12.2. Extended interrupts and events controller (EXTI)

The extended interrupt and event controller, through configurable (configurable) and direct (direct event) input (Lines), manages the CPU and system wake-up functions, and outputs the following request signals:

- Interrupt request, sent to the int_ctrl module to generate the IRQ of the CPU
- Event request, event input to CPU (RXEV)
- Wake-up request, sent to power management control module

EXTI wakeup request allows the system to wake up from stop mode, interrupt request and event request can also be used in run mode.

EXTI allows to manage up to 21 configurable/direct event lines (19 configurable event lines and 2 direct event lines).

12.2.1. EXTI main features

- The system can wake up through GPIO and specified module (PVD/COMP/RTC/LPTIM) input events
- Configurable events (from I/O, or peripherals with no state pending bits, peripherals that generate pulses)
  - ✓ Optional valid trigger edge (rising edge/falling edge)
  - ✓ Interrupt pending flag
  - ✓ Independent interrupt and event generation mask bit
  - ✓ Triggered by software
- Direct events (peripherals with associated flags and interrupt pending status bits)
  - ✓ Fixed rising edge trigger
  - ✓ No interrupt pending bit in EXTI module
  - ✓ Independent interrupt and event generation mask bit
  - ✓ No software trigger
- IO port selection

12.2.2. EXTI diagram

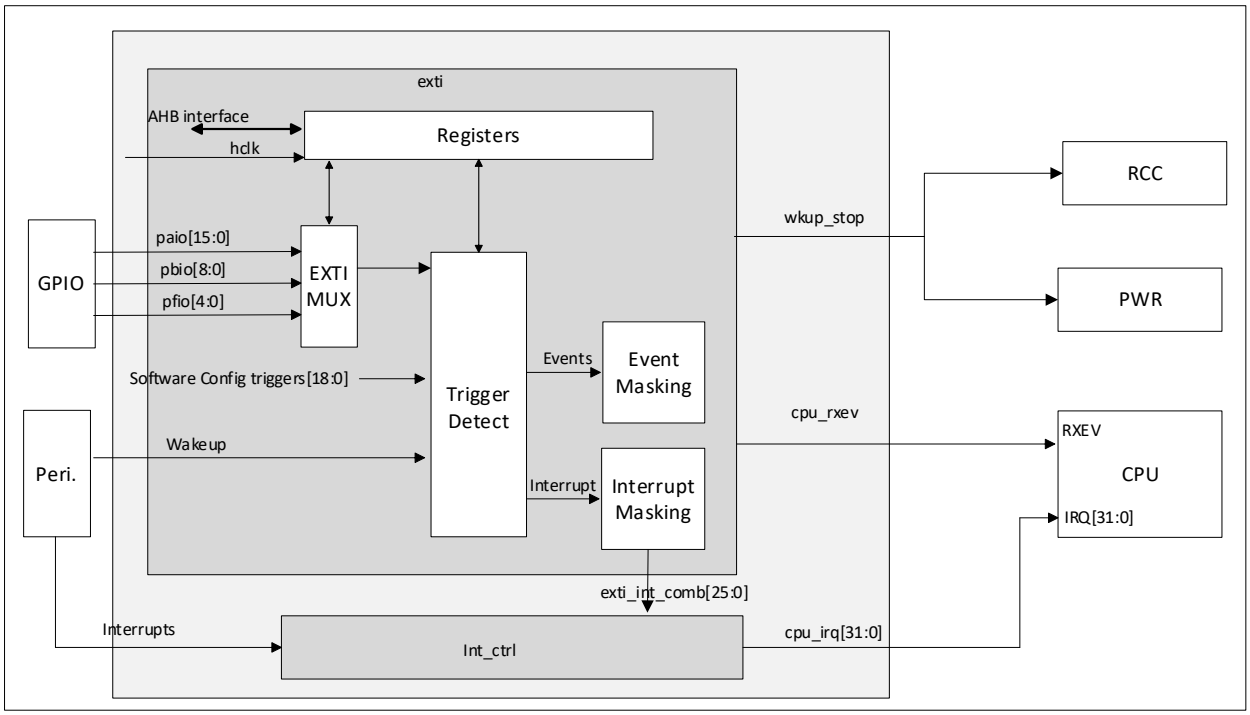


Figure 12-1 EXTI diagram

12.2.3. EXTI connection between peripherals and CPU

A peripheral that can generate a wake-up or interrupt event signal in stop mode is connected to the EXTI module.

- A wake-up signal that generates a pulse, or has no interrupt status bits inside the peripheral, is connected to the configurable line of the EXTI module. At this time, the EXTI module generates an interrupt pending bit (this bit needs to be cleared), and the EXTI interrupt will be used as the interrupt signal of the CPU.
- The interrupt and wake-up signal of the peripheral with the associated status bit (the bit is cleared in the peripheral) is connected to the wake-up trigger signal line of the EXTI module.
- All GPIO ports are input to the EXTI MUX module, and can be selected as a system wake-up signal through configurable configuration.

12.2.4. EXTI configurable event trigger wake-up

By configuring the EXTI_SWIER1 register, software can trigger the wake-up function.

There is a corresponding register configuration that triggers a rising edge or falling edge or a double edge to trigger a configurable type event. The hardware detects the input signal of the configurable type event according to the configuration, and generates a corresponding wake-up event or interrupt signal.

The CPU has dedicated interrupt mask registers and event mask registers. The event generated to the CPU after the event is enabled. The only event input signal rxev that is output to the CPU after all events to the CPU are OR'ed.

Configurable type events have a unique interrupt pending request register, which is shared with the CPU. The pending register is only set when the CPU Interrupt Mask Register (EXTI_IMR) is configured as unmasked. Each configurable type event corresponds to a CPU external interrupt signal (some will be multiplexed to the same CPU external interrupt signal). Configurable type event interrupt requires the CPU to confirm through the EXTI_PR register (write 1 to clear).

Note: When a bit of the interrupt pending register (EXTI_PR) remains valid (not cleared), the system cannot enter the low power consumption mode.

12.2.5. EXTI direct type event input wakeup

The direct type event will generate an interrupt in the EXTI module, and will generate an event signal to wake up the system and the CPU subsystem. When the CPU processes the interrupt generated by this type of trigger event, it needs to clear the interrupt status bit of the peripheral module.

12.2.6. External and internal interrupt/event line mapping

The GPIOs are connected to the 16 external interrupt/event lines in the following manner:

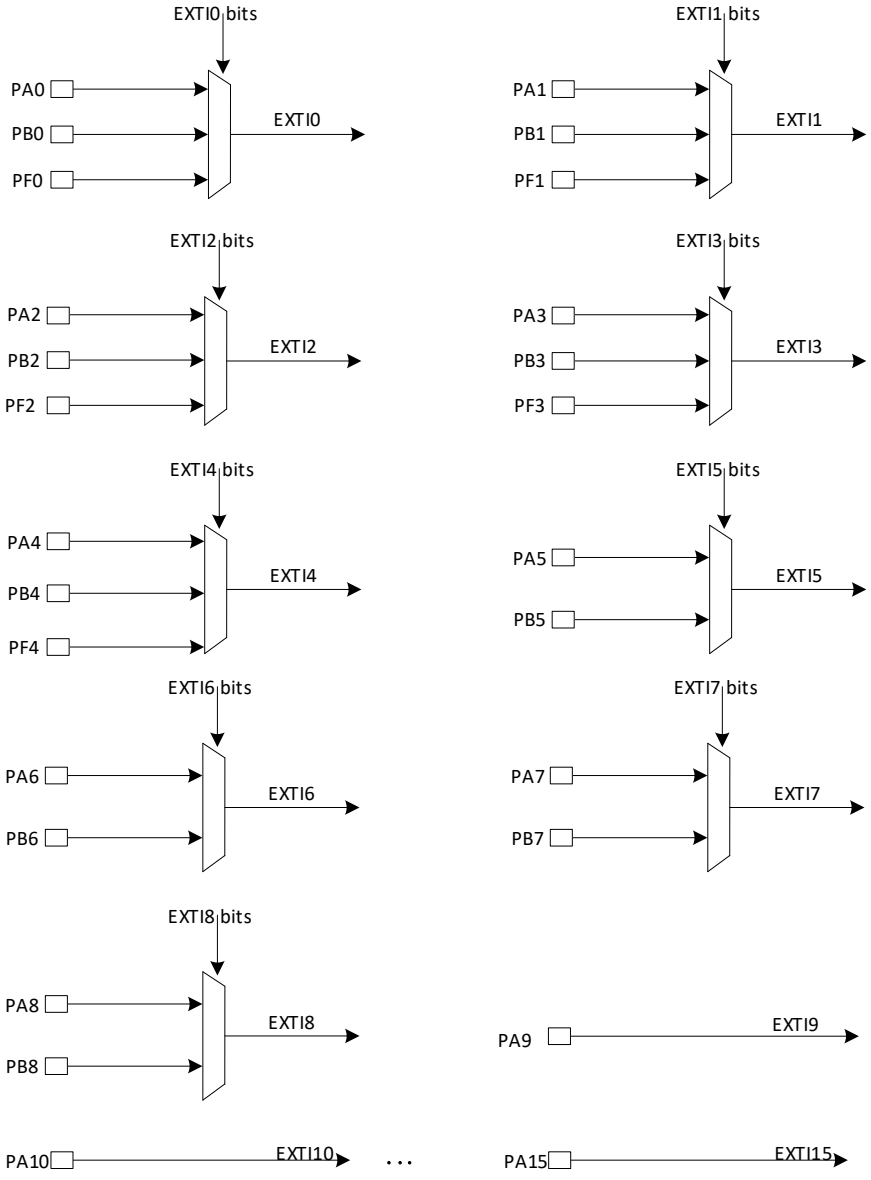


Figure 12-2 External interrupt/event GPIO mapping

The remaining lines are connected as follow:

EXTI line	Line source	Line type
Line 0-15	GPIO	configurable
Line 16	PVD output	Configurable
Line 17	COMP 1 output	Configurable
Line 18	COMP 2 output	Configurable
Line 19	RTC	Direct
Line 20	Reserved	

Line 21	Reserved	
Line 22	Reserved	
Line 23	Reserved	
Line 24	Reserved	
Line 25	Reserved	
Line 26	Reserved	
Line 27	Reserved	
Line 28	Reserved	
Line 29	LPTIM	Direct

12.3. EXTI registers

The registers of this peripheral can be accessed with word (32bit), half-word (16bit) and byte (8bit).

12.3.1. Rising trigger selection register (EXTI_RTSR)

Address offset: 0x00

Reset value: 0x0000 0000

Contains only register control bits for configurable events.

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		RT18	RT17	RT16
													RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved			
18	RT18	RW	0	Configurable type EXTI line18 rising edge trigger configuration. 0: Disable 1: enable
17	RT17	RW	0	Configurable type EXTI line17 rising edge trigger configuration. 0: Disable 1: enable
16	RT16	RW	0	Configurable type EXTI line16 rising edge trigger configuration. 0: Disable 1: enable
15	RT15	RW	0	Configurable type EXTI line15 rising edge trigger configuration. 0: Disable 1: enable
14	RT14	RW	0	Configurable type EXTI line14 rising edge trigger configuration. 0: Disable 1: enable
13	RT13	RW	0	Configurable type EXTI line13 rising edge trigger configuration. 0: Disable 1: enable
12	RT12	RW	0	Configurable type EXTI line12 rising edge trigger configuration. 0: Disable 1: enable
11	RT11	RW	0	Configurable type EXTI line11 rising edge trigger configuration. 0: Disable 1: enable
10	RT10	RW	0	Configurable type EXTI line10 rising edge trigger configuration. 0: Disable 1: enable
9	RT9	RW	0	Configurable type EXTI line9 rising edge trigger configuration. 0: Disable 1: enable
8	RT8	RW	0	Configurable type EXTI line8 rising edge trigger configuration. 0: Disable 1: enable

7	RT7	RW	0	Configurable type EXTI line7 rising edge trigger configuration. 0: Disable 1: enable
6	RT6	RW	0	Configurable type EXTI line6 rising edge trigger configuration. 0: Disable 1: enable
5	RT5	RW	0	Configurable type EXTI line5 rising edge trigger configuration. 0: Disable 1: enable
4	RT4	RW	0	Configurable type EXTI line4 rising edge trigger configuration. 0: Disable 1: enable
3	RT3	RW	0	Configurable type EXTI line3 rising edge trigger configuration. 0: Disable 1: enable
2	RT2	RW	0	Configurable type EXTI line2 rising edge trigger configuration. 0: Disable 1: enable
1	RT1	RW	0	Configurable type EXTI line1 rising edge trigger configuration. 0: Disable 1: enable
0	RT0	RW	0	Configurable type EXTI line0 rising edge trigger configuration. 0: Disable 1: enable

Configurable lines are edge-triggered, and glitches cannot be generated on these lines. If a rising edge occurs on the configurable interrupt line during a write to the EXTI_RTSR register, the associated Pending bit is not set. Both rising and falling edges can be set on the same line, in which case both edges will generate a trigger condition.

12.3.2. Falling trigger selection register (EXTI_FTSR)

Address offset: 0x04

Reset value: 0x0000 0000

Contains only register control bits for configurable events.

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	FT18	FT17	FT16
													RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved		-	
18	FT18	RW	0	Configurable type EXTI line18 falling edge trigger configuration. 0: Disable 1: enable
17	FT17	RW	0	Configurable type EXTI line17 falling edge trigger configuration. 0: Disable 1: enable
16	FT16	RW	0	Configurable type EXTI line16 falling edge trigger configuration. 0: Disable 1: enable
15	FT15	RW	0	Configurable type EXTI line15 falling edge trigger configuration. 0: Disable 1: enable
14	FT14	RW	0	Configurable type EXTI line14 falling edge trigger configuration. 0: Disable 1: enable
13	FT13	RW	0	Configurable type EXTI line13 falling edge trigger configuration. 0: Disable 1: enable
12	FT12	RW	0	Configurable type EXTI line12 falling edge trigger configuration. 0: Disable

				1: enable
11	FT11	RW	0	Configurable type EXTI line11 falling edge trigger configuration. 0: Disable 1: enable
10	FT10	RW	0	Configurable type EXTI line10 falling edge trigger configuration. 0: Disable 1: enable
9	FT9	RW	0	Configurable type EXTI line9 falling edge trigger configuration. 0: Disable 1: enable
8	FT8	RW	0	Configurable type EXTI line8 falling edge trigger configuration. 0: Disable 1: enable
7	FT7	RW	0	Configurable type EXTI line7 falling edge trigger configuration. 0: Disable 1: enable
6	FT6	RW	0	Configurable type EXTI line6 falling edge trigger configuration. 0: Disable 1: enable
5	FT5	RW	0	Configurable type EXTI line5 falling edge trigger configuration. 0: Disable 1: enable
4	FT4	RW	0	Configurable type EXTI line4 falling edge trigger configuration. 0: Disable 1: enable
3	FT3	RW	0	Configurable type EXTI line3 falling edge trigger configuration. 0: Disable 1: enable
2	FT2	RW	0	Configurable type EXTI line2 falling edge trigger configuration. 0: Disable 1: enable
1	FT1	RW	0	Configurable type EXTI line1 falling edge trigger configuration. 0: Disable 1: enable
0	FT0	RW	0	Configurable type EXTI line0 falling edge trigger configuration. 0: Disable 1: enable

Note: The external wakeup lines are edge triggered. No glitches must be generated on these lines. If a falling edge on an external interrupt line occurs during a write operation to the EXTI_FTSR register, the pending bit is not set.

Rising and falling edge triggers can be set for the same interrupt line. In this case, both generate a trigger condition.

12.3.3. Software interrupt event register (EXTI_SWIER)

Address offset: 0x08

Reset value: 0x0000 0000

Contains only register control bits for configurable events.

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	SW18	SW17	SW16
													RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SW15	SW14	SW13	SW12	SW11	SW10	SW9	SW8	SW7	SW6	SW5	SW4	SW3	SW2	SW1	SW0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:19	Reserved		-	
18	SWI18	RW	0	Configurable type EXTI line18 software rising edge trigger configuration. 0: No effect

				1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
17	SWI17	RW	0	Configurable type EXTI line17 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
16	SWI16	RW	0	Configurable type EXTI line16 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
15	SWI15	RW	0	Configurable type EXTI line15 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
14	SWI14	RW	0	Configurable type EXTI line14 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
13	SWI13	RW	0	Configurable type EXTI line13 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
12	SWI12	RW	0	Configurable type EXTI line12 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
11	SWI11	RW	0	Configurable type EXTI line11 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
10	SWI10	RW	0	Configurable type EXTI line10 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
9	SWI9	RW	0	Configurable type EXTI line9 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
8	SWI8	RW	0	Configurable type EXTI line8 software rising edge trigger configuration. 0: No effect

				1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
7	SWI7	RW	0	Configurable type EXTI line7 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
6	SWI6	RW	0	Configurable type EXTI line6 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
5	SWI5	RW	0	Configurable type EXTI line5 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
4	SWI4	RW	0	Configurable type EXTI line4 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
3	SWI3	RW	0	Configurable type EXTI line3 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
2	SWI2	RW	0	Configurable type EXTI line2 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
1	SWI1	RW	0	Configurable type EXTI line1 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)
0	SWI0	RW	0	Configurable type EXTI line0 software rising edge trigger configuration. 0: No effect 1: Generate a rising edge trigger event, which in turn generates an interrupt This bit is cleared by hardware, and a read returns 0 (after hardware clearing) or configuration value (before hardware clearing)

12.3.4. Pending register (EXTI_PR)

Address offset: 0x0C

Reset value: undefined

Contains only register control bits for configurable events.

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	PR18	PR17	PR16
													rc_w1	rc_w1	rc_w1
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

Bit	Name	R/W	Reset Value	Function
31:19	Reserved	reserved	-	
18	PR18	RC_W1	0	Configurable type EXTI line18 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
17	PR17	RC_W1	0	Configurable type EXTI line17 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
16	PR16	RC_W1	0	Configurable type EXTI line16 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
15	PR15	RC_W1	0	Configurable type EXTI line15 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
14	PR14	RC_W1	0	Configurable type EXTI line14 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
13	PR13	RC_W1	0	Configurable type EXTI line13 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
12	PR12	RC_W1	0	Configurable type EXTI line12 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
11	PR11	RC_W1	0	Configurable type EXTI line11 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,

10	PR10	RC_W1	0	Configurable type EXTI line10 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
9	PR9	RC_W1	0	Configurable type EXTI line9 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
8	PR8	RC_W1	0	Configurable type EXTI line8 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
7	PR7	RC_W1	0	Configurable type EXTI line7 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
6	PR6	RC_W1	0	Configurable type EXTI line6 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
5	PR5	RC_W1	0	Configurable type EXTI line5 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
4	PR4	RC_W1	0	Configurable type EXTI line4 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
3	PR3	RC_W1	0	Configurable type EXTI line3 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
2	PR2	RC_W1	0	Configurable type EXTI line2 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,
1	PR1	RC_W1	0	Configurable type EXTI line1 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,

0	PR0	RC_W1	0	Configurable type EXTI line0 event pending flag. This bit is set when software or hardware generates a rising/falling edge trigger event. Software writes 1 to clear. 0: no event request is generated, 1: Generate rising edge/falling edge/software trigger event request,

12.3.5. External interrupt select register 1 (EXTI_EXTICR1)

Address offset: 0x60

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	EXTI3[1:0]		Res	Res	Res	Res	Res	Res	EXTI2[1:0]	
						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	EXTI1[1:0]		Res	Res	Res	Res	Res	Res	EXTI0[1:0]	
						RW	RW							RW	RW

Bit	Name	R/W	Reset Value	Function
31:21	Reserved	-	-	Reserved
25:24	EXTI3[1:0]	RW	0	EXTI3 corresponds to GPIO port selection. 2'b00: PA[3] pin 2'b01: PB[3] pin 2'b10: PF[3] pin 2'b11: reserved
23:18	Reserved	-	-	Reserved
17:16	EXTI2[1:0]	RW	0	EXTI2 corresponds to GPIO port selection. 2'b00: PA[2] pin 2'b01: PB[2] pin 2'b10: PF[2] pin 2'b11: reserved
15:10	Reserved	-	-	Reserved
9:8	EXTI1[1:0]	RW	0	EXTI1 corresponds to GPIO port selection. 2'b00: PA[1] pin 2'b01: PB[1] pin 2'b10: PF[1] pin 2'b11: reserved
7:2	Reserved	-	-	Reserved
1:0	EXTI0[1:0]	RW	0	EXTI0 corresponds to GPIO port selection. 2'b00: PA[0] pin 2'b01: PB[0] pin 2'b10: PF[0] pin 2'b11: reserved

12.3.6. External interrupt select register 2 (EXTI_EXTICR2)

Address offset: 0x64

Reset value: 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	EXTI7	Res	Res	Res	Res	Res	Res	Res	EXTI6
							RW								RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	Res	Res	Res	RW	EXTI5	Res	Res	Res	Res	Res	Res	EXTI4[1:0]	
							RW							RW	RW

Bit	Name	R/W	Reset Value	Function
31:20	Reserved	-	-	Reserved
19	EXTI7	RW	0	EXTI7 corresponds to GPIO port selection. 0: PA[7] pin 1: PB[7] pin
23:18	Reserved	-	-	Reserved
17:16	EXTI6	RW	0	EXTI6 corresponds to GPIO port selection. 0: PA[6] pin

				1: PB[6] pin
15:9	Reserved	-	-	Reserved
8	EXTI5	RW	0	EXTI5 corresponds to GPIO port selection. 0: PA[5] pin 1: PB[5] pin
7:2	Reserved	-	-	Reserved
1:0	EXTI4[1:0]	RW	0	EXTI4 corresponds to GPIO port selection. 2'b00: PA[4] pin 2'b01: PB[4] pin 2'b10: PF[4] pin 2'b11: reserved

12.3.7. External interrupt select register 3 (EXTI_EXTICR3)

Address offset: 0x68

Reset value: 0x0000

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	Res	Res	Res	Res	Res	Res	Res	Res	EXTI8
															RW

Bit	Name	R/W	Reset Value	Function
31:1	Reserved	-	-	Reserved
0	EXTI8	RW	0	EXTI8 corresponds to GPIO port selection. 0: PA[8] pin 1: PB[8] pin

12.3.8. Interrupt mask register (EXTI_IMR)

Address offset: 0x80

Reset value: 0x2008 0000

Note: The interrupt mask bit of the Direct type line is 1 by default, that is, the line is not masked, the mask bit of the configurable line, the default is 0, that is, the line is masked.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	IM29	Res	Res	Res	Res	Res	Res	Res	Res	Res	IM19	IM18	IM17	IM16
		RW										RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:30	Reserved			
29	IM29	RW	1	EXTI line29 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
28:20	Reserved			
19	IM19	RW	1	EXTI line19 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
18	IM18	RW	0	EXTI line18 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
17	IM17	RW	0	EXTI line17 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
16	IM16	RW	0	EXTI line16 is used as an interrupt to wake up the CPU mask control.

				0: interrupt wake-up mask 1: Interrupt wake-up is not masked
15	IM15	RW	0	EXTI line15 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
14	IM14	RW	0	EXTI line14 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
13	IM13	RW	0	EXTI line13 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
12	IM12	RW	0	EXTI line12 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
11	IM11	RW	0	EXTI line11 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
10	IM10	RW	0	EXTI line10 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
9	IM9	RW	0	EXTI line9 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
8	IM8	RW	0	EXTI line8 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
7	IM7	RW	0	EXTI line7 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
6	IM6	RW	0	EXTI line6 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
5	IM5	RW	0	EXTI line5 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
4	IM4	RW	0	EXTI line4 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
3	IM3	RW	0	EXTI line3 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
2	IM2	RW	0	EXTI line2 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
1	IM1	RW	0	EXTI line1 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked
0	IM0	RW	0	EXTI line0 is used as an interrupt to wake up the CPU mask control. 0: interrupt wake-up mask 1: Interrupt wake-up is not masked

12.3.9. Event mask register (EXTI_EMR)

Address offset: 0x84

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	EM29	Res	Res	Res	Res	Res	Res	Res	Res	Res	EM19	EM18	EM17	EM16
		RW										RW	RW	RW	RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:30	Reserved			
29	EM29	RW	0	EXTI line29 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
28:20	Reserved			
19	EM19	RW	0	EXTI line19 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
18	EM18	RW	0	EXTI line18 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
17	EM17	RW	0	EXTI line17 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
16	EM16	RW	0	EXTI line16 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
15	EM15	RW	0	EXTI line15 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
14	EM14	RW	0	EXTI line14 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
13	EM13	RW	0	EXTI line13 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
12	EM12	RW	0	EXTI line12 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
11	EM11	RW	0	EXTI line11 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
10	EM10	RW	0	EXTI line10 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
9	EM9	RW	0	EXTI line9 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
8	EM8	RW	0	EXTI line8 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked

7	EM7	RW	0	EXTI line7 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
6	EM6	RW	0	EXTI line6 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
5	EM5	RW	0	EXTI line5 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
4	EM4	RW	0	EXTI line4 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
3	EM3	RW	0	EXTI line3 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
2	EM2	RW	0	EXTI line2 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
1	EM1	RW	0	EXTI line1 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked
0	EM0	RW	0	EXTI line0 wakes up the CPU mask control as an event. 0: Event wake-up mask 1: Event wakeup is not masked

12.3.10. EXTI register map

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
0x00	EXTI_RTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT18	RT17	RT16	RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	EXTI_FTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT18	FT17	FT16	FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	EXTI_SWIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI18	SWI17	SWI16	SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	EXTI_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR18	PR17	PR16	PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10-0x	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.

0x5C																																	
0x60	EXTI_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI2[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	EXTI1[1:0]		Res.	Res.	Res.	Res.	Res.	EXTI0[1:0]	
	Reset value							0	0								0	0							0	0						0	0
0x64	EXTI_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI5	Res.	Res.	Res.	Res.	Res.	Res.	EXTI4[1:0]
	Reset value									0								0									0						0	0
0x68	EXTI_EXTICR3	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.	EXTI8
	Reset value																																	0
0x6C-0x7C	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.	Res.
0x80	EXTI_IMR	Res.	Res.	Res.	IM29	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM19	IM18	IM17	IM16	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
	Reset value				1												1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	EXTI_EMR	Res.	Res.	Res.	EM29	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM19	EM18	EM17	EM16	EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
	Reset value				0												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13. Cyclic redundancy check calculation unit (CRC)

13.1. Introduction

According to the generator polynomial, the CRC calculation unit will operate the input 32-bit data to generate a CRC result.

13.2. CRC main features

- Uses CRC-32 (Ethernet) polynomial: 0x4C11DB7

  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- Support 32-bit data input
- A single input/output 32 data and result output share one register
- 8-bit register for general purpose (can be used as temporary storage)
- Computation time: 4 AHB clocks for 32bits data

13.3. CRC functional description

13.3.1. CRC block diagram

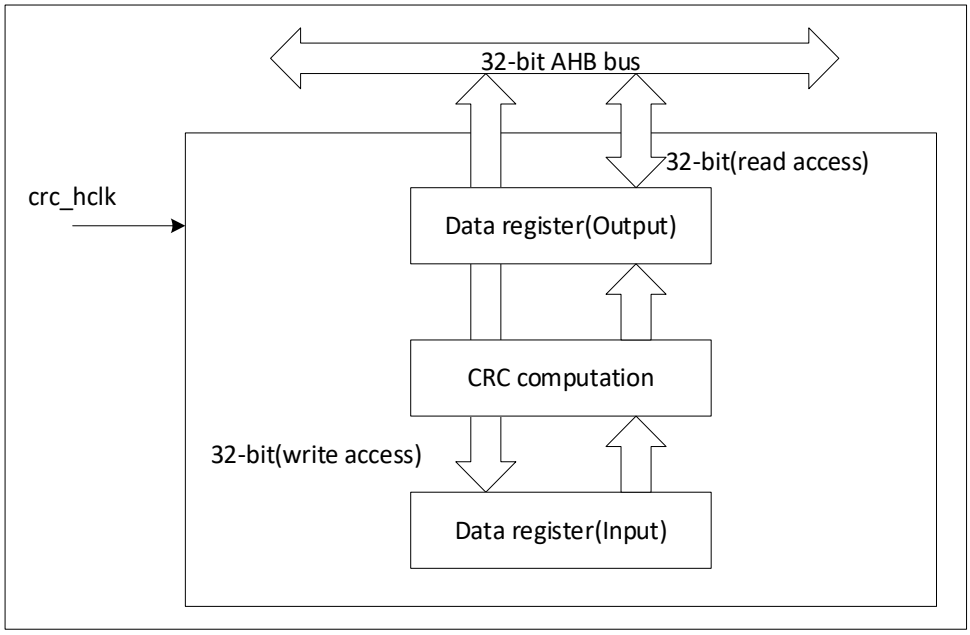


Figure 13-1 CRC calculation unit block diagram

The CRC calculation unit contains a 32-bit data register:

- When writing to this register, as an input register, new data to be calculated by CRC can be input.
- When the register is read, the result of the last CRC calculation is returned.

Each time a data register is written, the result of the calculation is the combination of the previous CRC calculation and the new calculation (CRC is calculated on the entire 32-bit word, not byte by byte).

While the CRC is being calculated, writes are blocked until the end of the CRC calculation.

The register CRC_DR can be reset to 0xFFFF FFFF by setting the RESET bit of the register CRC_CR. This operation does not affect the data in register CRC_IDR.

13.4. CRC registers

13.4.1. Data register (CRC_DR)

Address offset: 0x00

Reset value: 0xFFFF FFFF

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

Bit	Name	R/W	Reset Value	Function
31:0	DR	RW	32'hFFFFFFFF	data register. When writing new data, it is used as an input register. When read, the previous CRC calculation result is retained.

13.4.2. Independent data register (CRC_IDR)

Address offset: 0x04

Reset value: 0x0000 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	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	Res	IDR[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7:0	IDR[7:0]	RW	0	General purpose 8bit data register These bits are used as temporary storage for one byte. This register is not reset by the RESET bit of the CRC_CR register.

13.4.3. Control register (CRC_CR)

Address offset: 0x08

Reset value: 0x0000 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	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	Res	Res	Res	Res	Res	Res	Res	Res	RESET
															W

Bit	Name	R/W	Reset Value	Function
31:1	Reserved		-	
0	RESET		0	This bit is set by software to reset the CRC calculation unit. This bit can only be set and is automatically cleared by hardware.

13.4.4. CRC register map

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
0x00	CRC_DR	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	CRC_IDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x08	CRC_CR	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.	RESET
	Reset value																																0

14. Analog-to-digital converter (ADC)

14.1. Introduction

The chip has a 12-bit SARADC (successive approximation analog-to-digital converter). The module has a total of 12 channels to be measured, including 10 external channels and 2 internal channels.

The conversion mode of each channel can be set to single, continuous, sweep, discontinuous mode.Conversion results are stored in left- or right-aligned 16-bit data registers.

The analog watchdog feature allows the application to detect if the input voltage goes outside the user-defined higher or lower thresholds.

An efficient low-power mode is implemented to allow very low consumption at low frequency.

14.2. ADC main features

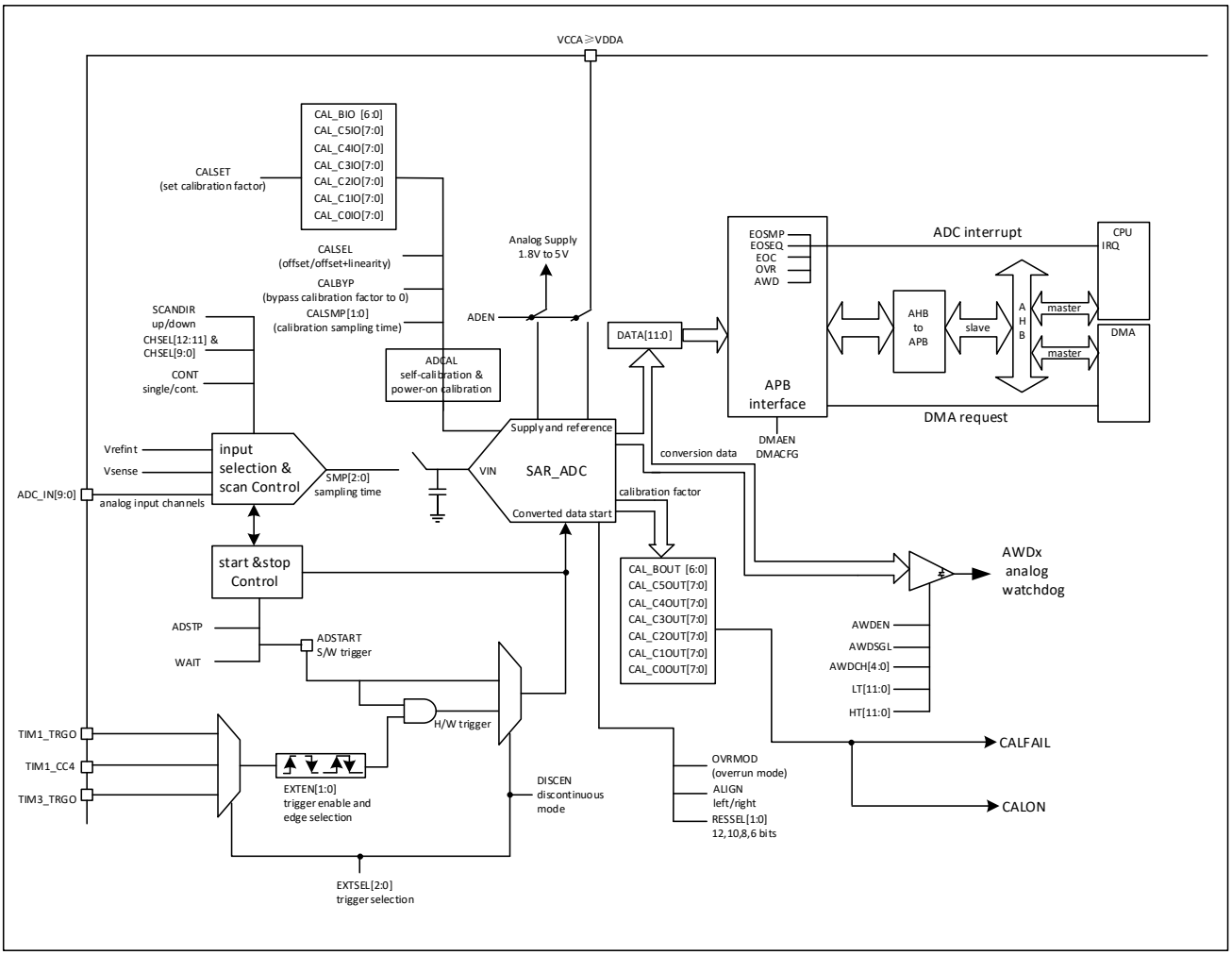
- High performance
  - 12-bit, 10-bit, 8-bit or 6-bit configurable resolution
  - ADC conversion time: 1.0 μs for 12-bit resolution (1 MHz)
  - Self-calibration
  - Programmable sampling time
  - Programmable data alignment mode
  - DMA support
- Low-power
  - Application can reduce PCLK frequency for low-power operation while still keeping optimum ADC performance.
  - Wait mode: prevents ADC overrun in applications with low frequency PCLK
- Analog input channels
  - 10 external analog inputs:PA[7:0] and PB[1:0]
  - 1 channel for internal temperature sensor (VSENSE)
  - 1 channel for internal reference voltage (VREFINT)
- Start-of-conversion can be initiated:
  - By software
  - By hardware triggers with configurable polarity (internal timer events from TIM1, TIM3 and GPIO)
- Conversion modes
  - Single mode: Can convert a single channel or can scan a sequence of channels
  - Continuous mode: Continuous mode converts selected inputs continuously
  - Discontinuous mode: Convert selected channel once per trigger
- Interrupt generation
  - At the end of sampling
  - At the end of conversion

- At the end of sequence conversion
- In case of analog watchdog
- Overrun events

■ Analog watchdog

14.3. ADC functional description

14.3.1. ADC diagram



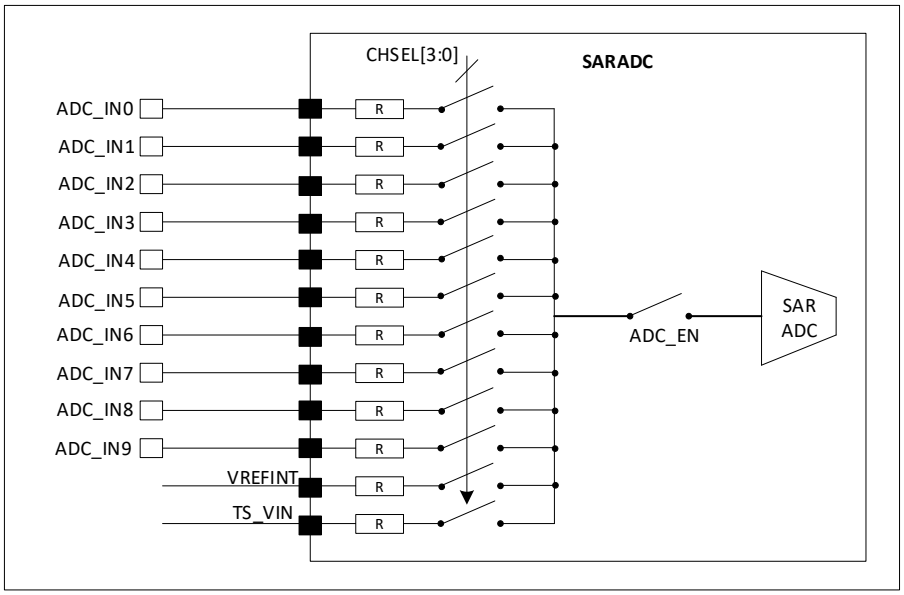


Figure 14-1 ADC channel with analog switch

14.3.2. Calibration (ADCAL)

The ADC has a calibration feature. During the procedure, the ADC calculates a calibration factor which is internally applied to the ADC until the next ADC power-off. The application must not use the ADC during calibration and must wait until it is complete.

Calibration operations include power-on calibration and software calibration.

ADC power-on calibration

The hardware will automatically perform ADC calibration after power-up.

ADC software calibration

The software can set ADCAL = 1 to start the calibration. The calibration can only be started when the ADC is not enabled (ADEN = 0), and only the system clock can be selected as the ADC clock. ADCAL is cleared by hardware when calibration is complete.

When the working conditions of the ADC change (the change in VCC is the main factor for the offset of the ADC, followed by the change in temperature), it is recommended to perform a re-calibration operation.

Calibration software procedure:

- Ensure that ADEN = 0、CKMODE selects the system clock
- Set ADCAL = 1
- Wait until ADCAL = 0

14.3.3. ADC on-off control (ADEN)

At MCU power-up, the ADC is disabled and put in power-down mode (ADEN = 0).

The following is the process to enable ADC:

1. Write 1 to clear the ADRDY bit in the ADC_ISR register
2. The ADEN bit of the ADC_CR register is set to 1

ADC conversions are also initiated by setting ADSRART or (if triggered) by an external trigger event.

The following is the procedure for disabling the ADC:

Check that ADSTART in the ADC_CR register is 0 to ensure the ADC is not in the process of converting. If necessary, set ADSTP in the ADC_CR register to 1 to stop the ongoing ADC conversion, and wait for ADSTP to be cleared by hardware (cleared to 0 means the conversion is stopped).

Warning: ADEN bit cannot be set to 1 during 4 ADC clocks after ADCAL is cleared by hardware and ADCAL = 1.

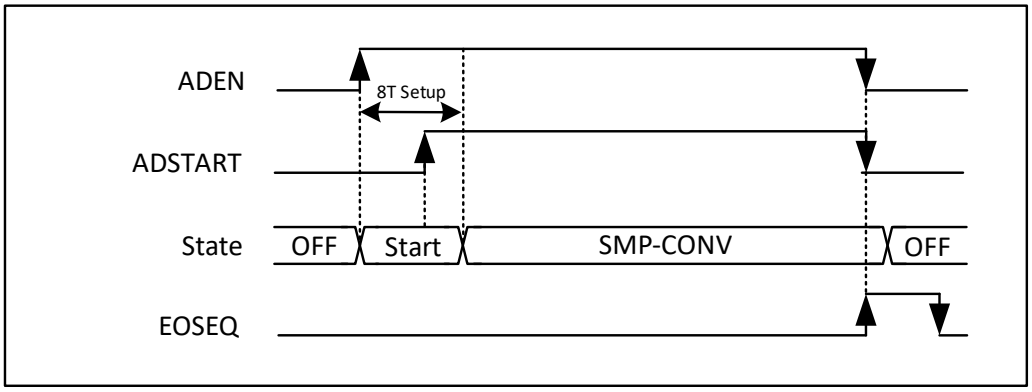


Figure 14-2 Enabling/disabling the ADC

14.3.4. ADC click

The ADC has a dual clock-domain architecture, so that the ADC can be fed with a clock (ADC_CLK) independent from the APB clock (PCLK). ADC_CLK can be generated by two possible clock sources.

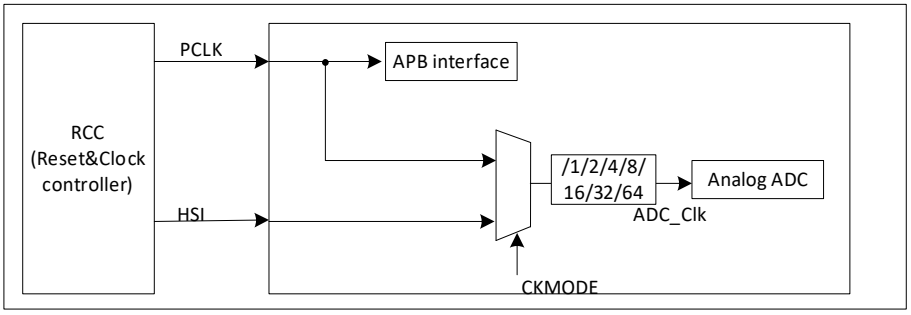


Figure 14-3 ADC clock scheme

Table 14-1 Delay between trigger and conversion start

ADC clock source	CKMODE[3:0]	Frequency division factor	Latency between the trigger event and the start of conversion (T is the clock period)
PCLK	0000	1	0
	0001	2	0
	0010	4	0
	0011	8	0
	0100	16	0
	0101	32	0
	0110	64	0
	0111	/	/
HSI	1000	1	0
	1001	2	0
	1010	4	0
	1011	8	0
	1100	16	0
	1101	32	0
	1110	64	0
	1111	/	/

14.3.5. Configuring the ADC

Software must write to the ADCAL and ADEN bits in the ADC_CR register if the ADC is disabled (ADEN must be 0).

Software must only write to the ADSTART and ADDIS bits in the ADC_CR register only if the ADC is enabled and there is no pending request to disable the ADC (ADEN = 1 and ADDIS = 0).

For all the other control bits in the ADC_IER, ADC_CFGRi, ADC_SMPR, ADC_TR, ADC_CHSELR and ADC_CCR registers, software must only write to the configuration control bits if the ADC is enabled (ADEN = 1) and if there is no conversion ongoing (ADSTART = 0).

Software must only write to the ADSTP bit in the ADC_CR register if the ADC is enabled (and possibly converting) and there is no pending request to disable the ADC (ADSTART = 1 and ADDIS = 0).

14.3.6. Channel selection (CHSEL, SCANDIR)

There are up to 12 multiplexed channels:

● 10 analog inputs from GPIO pins (ADC_IN0...ADC_IN9)

● 2 internal analog inputs (Temperature Sensor, Internal Reference Voltage)

It is possible to convert a single channel or to automatically scan a sequence of channels.

The sequence of the channels to be converted must be programmed in the ADC_CHSELR channel selection register: each analog input channel has a dedicated selection bit (CHSEL0...CHSEL11).

The order in which the channels will be scanned can be configured by programming the bit SCANDIR bit in the ADC_CFGR1 register:

● SCANDIR = 0: foRWard scan Channel 0 to Channel 11

● SCANDIR = 1: backward scan Channel 11 to Channel 0

Temperature sensor, VREFINT internal channels

The temperature sensor is connected to channel ADC_IN10. The internal voltage reference VREFINT is connected to channel ADC_IN11.

14.3.7. Programmable sampling time (SMP)

Before starting a conversion, the ADC needs to establish a direct connection between the voltage source to be measured and the embedded sampling capacitor of the ADC. This sampling time must be enough for the input voltage source to charge the sample and hold capacitor to the input voltage level.

Having a programmable sampling time allows to trim the conversion speed according to the input resistance of the input voltage source.

The ADC samples the input voltage for a number of ADC clock cycles that can be modified using the SMP[2:0] bits in the ADC_SMPR register.

This programmable sampling time is common to all channels. If required by the application, the software can change and adapt this sampling time between each conversions.

The total conversion time is calculated as follows:

$$t_{CONV}$$ = Sampling time + (Convert resolution +0.5) x ADC clock cycles

Example:

When ADC_CLK = 16MHz, the resolution is 12 bits, and the sampling time is 3.5 ADC clock cycles:

t_{CONV} = (3.5 + 12.5) x ADC clock period = 16 x ADC clock period = 1 µs

The ADC indicates the end of the sampling phase by setting the EOSMP flag.

14.3.8. Single conversion mode (CONT = 0, DISCEN = 0)

In Single conversion mode, the ADC performs a single sequence of conversions, converting all the channels once. This mode is selected when CONT = 0, DISCEN = 0 in the ADC_CFGR1 register.

ADC conversions can be initiated in two ways:

● Set ADSTART bit in ADC_CR register

● Hardware trigger events

Inside the sequence, after each conversion is complete:

- The converted data are stored in the 16-bit ADC_DR register
- The EOC (end of conversion) flag is set
- An interrupt is generated if the EOCIE bit is set

After the sequence of conversions is complete:

- The EOSEQ (end of sequence) flag is set
- An interrupt is generated if the EOSEQIE bit is set

Then the ADC stops until a new external trigger event occurs or the ADSTART bit is set again.

Note: To convert a single channel, program a sequence with a length of 1.

The ADC cannot be in discontinuous conversion mode and continuous conversion mode at the same time, in this case (DISCEN = 1, CONT = 1), it behaves as single conversion mode.

14.3.9. Continuous conversion mode (CONT = 1)

In continuous conversion mode, when a software or hardware trigger event occurs, the ADC performs a sequence of conversions, converting all the channels once and then automatically re-starts and continuously performs the same sequence of conversions. This mode is selected when CONT = 1 in the ADC_CFGR1 register.

Conversion is started by either:

- Setting the ADSTART bit in the ADC_CR register
- Hardware trigger event

Inside the sequence, after each conversion is complete:

- The converted data are stored in the 16-bit ADC_DR registe
- The EOC (end of conversion) flag is set
- An interrupt is generated if the EOCIE bit is set

After the sequence of conversions is complete:

- The EOSEQ (end of sequence) flag is set
- An interrupt is generated if the EOSEQIE bit is set

Then, a new sequence restarts immediately and the ADC continuously repeats the conversion sequence.

Note: To convert a single channel, program a sequence with a length of 1.

It is not possible to have both discontinuous mode and continuous mode enabled: it is forbidden to set both bits DISCEN = 1 and CONT = 1.

14.3.10. Discontinuous conversion mode (DISCEN = 1)

This mode is enabled by setting the DISCEN bit in the ADC_CFGR1 register.

In this mode (DISCEN = 1), a hardware or software trigger event is required to initiate each conversion defined in a sequence.

Conversely, when DISCEN = 0, a hardware or software trigger event can initiate all conversions defined in a sequence.

For example:

DISCEN = 1, the channels to be converted are: 0, 3, 7, 10

- 1st trigger: Channel 0 is converted and an EOC event occurs
- 2nd trigger: Channel 3 is converted and an EOC event occurs
- 3rd trigger: Channel 7 is converted and an EOC event occurs
- 4th trigger: Channel 10 is converted and EOC and EOSEQ events are generated
- 5th trigger: Channel 0 is converted and an EOC event occurs
- 6th trigger: Channel 3 is converted and an EOC event occurs
- ...

DISCEN = 0, the channels to be converted are: 0, 3, 7, 10

- 1st Trigger: The entire complete sequence of conversions, in turn, channels 0, 3, 7, and 10.

Each conversion is completed, an EOC event is generated, and the conversion to the last channel generates an EOSEQ event in addition to the EOC.

- Any trigger event restarts the complete sequence conversion.

Note: It is impossible to have the ADC in continuous mode and continuous conversion mode at the same time, in this case (DISCEN = 1, CONT = 1), it behaves as a single conversion mode.

14.3.11. Starting conversions (ADSTART)

Software starts ADC conversion with setting ADSTART = 1.

When ADSTART is set, the conversion:

- When EXTEN = 0x0 (software trigger), start immediately
- When if EXTEN ≠ 0x0, start at the next selected hardware trigger valid edge

The ADSTART bit is also used to indicate whether an ADC conversion operation is currently in progress. When ADSTART = 0, the ADC can be reconfigured, indicating that the ADC is idle at this time.

ADSTART bit can be cleared by hardware.

- One-shot conversion mode is triggered by software (CONT = 0, EXTSEL = 0x0)

- After sequence conversion is complete (EOSEQ = 1)

- Discontinuous conversion mode is triggered by software (CONT = 0, DISCEN = 1, EXTSEL = 0x0)

- After conversion is complete (EOC = 1)

- In all cases (CONT = X, EXTSEL = X)

- After the software calls and executes the ADSTP procedure

Note: In continuous mode (CONT = 1), the ADSTART bit cannot be cleared by hardware caused by EOSEQ because it automatically restarts the sequence conversion. When the hardware trigger is selected as single conversion mode (CONT = 0 and EXTSEL = 0x01), ADSTART will not be cleared by hardware after the EOSEQ

flag is set. This avoids the need for software to reset the ADSTART bit and ensures that no hardware trigger event is missed.

14.3.12. Timings

The elapsed time between the start of a conversion and the end of conversion is the sum of the configured sampling time plus the successive approximation time depending on data resolution:

$t_{ADC} = t_{SMPL} + t_{SAR} = [\ 3.5_{|min} + 12.5_{|12bit}] * t_{ADC\_CLK}$

$t_{ADC} = t_{SMPL} + t_{SAR} =\ 218.75ns_{|min} + 781.25\ ns_{|12bit} = 1\ \mu s_{|min}$ (for f_{ADC_CLK} = 16 MHz)

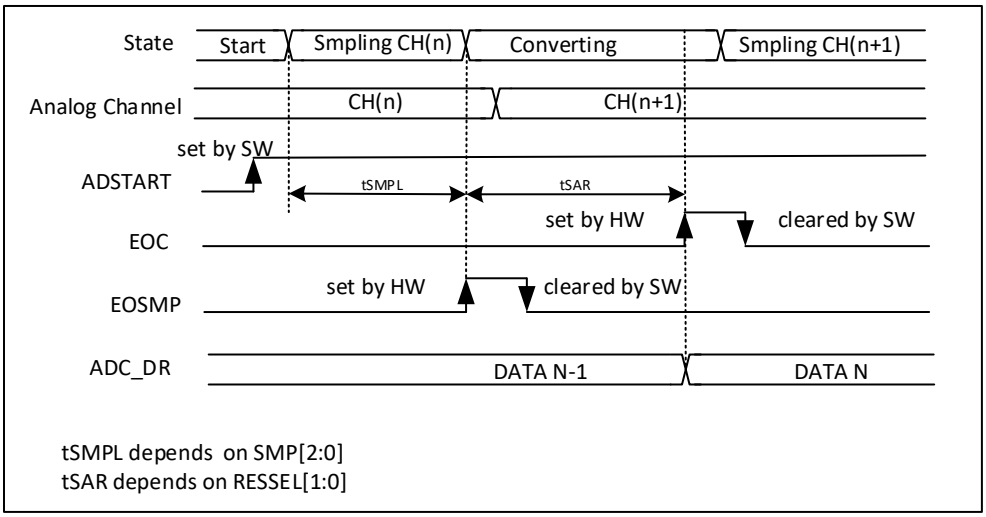


Figure 14-4 analog to digital conversion timing

14.3.13. Stopping an ongoing conversion (ADSTP)

The software can decide to stop any ongoing conversions by setting ADSTP = 1 in the ADC_CR register.

This will reset the ADC operation and the ADC will be idle, ready for a new operation.

When the ADSTP bit is set by software, any ongoing conversion is aborted and the result is discarded (ADC_DR register is not updated with the current conversion).

The scan sequence is also aborted and reset (meaning that restarting the ADC would restart a new sequence).

Once this procedure is complete, the ADSTP and ADSTART bits are both cleared by hardware.

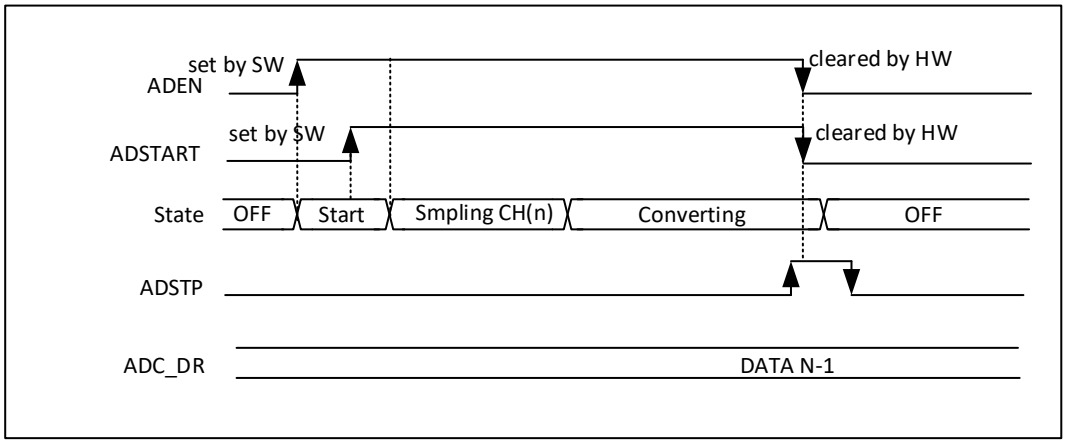


Figure 14-5 Stop timing

14.4. Conversion on external trigger and trigger polarity (EXTSEL, EXTEN)

A conversion or a sequence of conversion can be triggered either by software or by an external event (for example timer capture). If the EXTEN[1:0] control bits are not equal to “0b00”, then external events are able to trigger a conversion with the selected polarity. The trigger selection is effective once software has set bit ADSTART = 1.

Any hardware triggers which occur while a conversion is ongoing are ignored.

If bit ADSTART = 0, any hardware triggers which occur are ignored.

Source	EXTEN[1:0]
Trigger detection disabled	00
Detect on rising edge	01
Detect on falling edge	10
Detects on rising and falling edges	11

Note: The polarity of the external trigger can be changed only when the ADC is not converting (ASTART = 0).

The EXTSEL[2:0] control bits are used to select which of 8 possible events can trigger conversions.

The following table shows possible external triggers for rule transitions. A software source trigger event can be generated by setting the ADSTART bit in the ADC_CR register.

Table 14-2 External trigers

Name	Source	EXTSEL[2:0]
EXT0	TIM1_TRGO	000
EXT1	TIM1_CC4	001
EXT2	Reserved	010
EXT3	TIM3_TRGO	011
EXT4	Reserved	100
EXT5	Reserved	101
EXT6	Reserved	110

Note: The trigger selection can be changed only when the ADC is not converting (ADSTART = 0).

14.4.1. Programmable resolution (RES) - fast conversion mode

It is possible to obtain faster conversion times (tSAR) by reducing the ADC resolution.

The resolution can be configured to be either 12, 10, 8, or 6 bits by programming the RES[1:0] bits in the ADC_CFGR1 register. Lower resolution allows faster conversion times for applications where high data precision is not required.

Lower resolution mode reduces the conversion time of successive approximation as shown in the table below:

RESSEL [1:0]	t_{SAR} (ADC clock cycles)	t_{SAR}(ns) @ fADC = 24MHz	t_{SMP} (ADC clock cycles)	t_{ADC}(t_{SMP} = 3.5) (ADC clock cycles)	t_{CONV}(ns) @ f_{ADC} = 24MHz
12	12.5	521ns	3.5	16	667ns
10	10.5	438ns	3.5	14	583ns
8	8.5	396ns	3.5	12	500ns
6	6.5	271ns	3.5	10	417ns

14.4.2. End of conversion, end of sampling phase (EOC, EOSMP flags)

The ADC indicates each end of conversion (EOC) event.

The ADC sets the EOC flag in the ADC_ISR register as soon as a new conversion data result is available in the ADC_DR register. An interrupt can be generated if the EOCIE bit is set in the ADC_IER register. The EOC flag is cleared by software either by writing 1 to it, or by reading the ADC_DR register.

The ADC also indicates the end of sampling phase by setting the EOSMP flag in the ADC_ISR register. The EOSMP flag is cleared by software by writing1 to it. An interrupt can be generated if the EOSMPIE bit is set in the ADC_IER register.

14.4.3. End of conversion sequence (EOSEQ flag)

The ADC notifies the application of each end of sequence (EOSEQ) event.

The ADC sets the EOSEQ flag in the ADC_ISR register as soon as the last data result of a conversion sequence is available in the ADC_DR register. An interrupt can be generated if the EOSEQIE bit is set in the ADC_IER register. The EOSEQ flag is cleared by software by writing 1.

14.4.4. Example timing diagrams

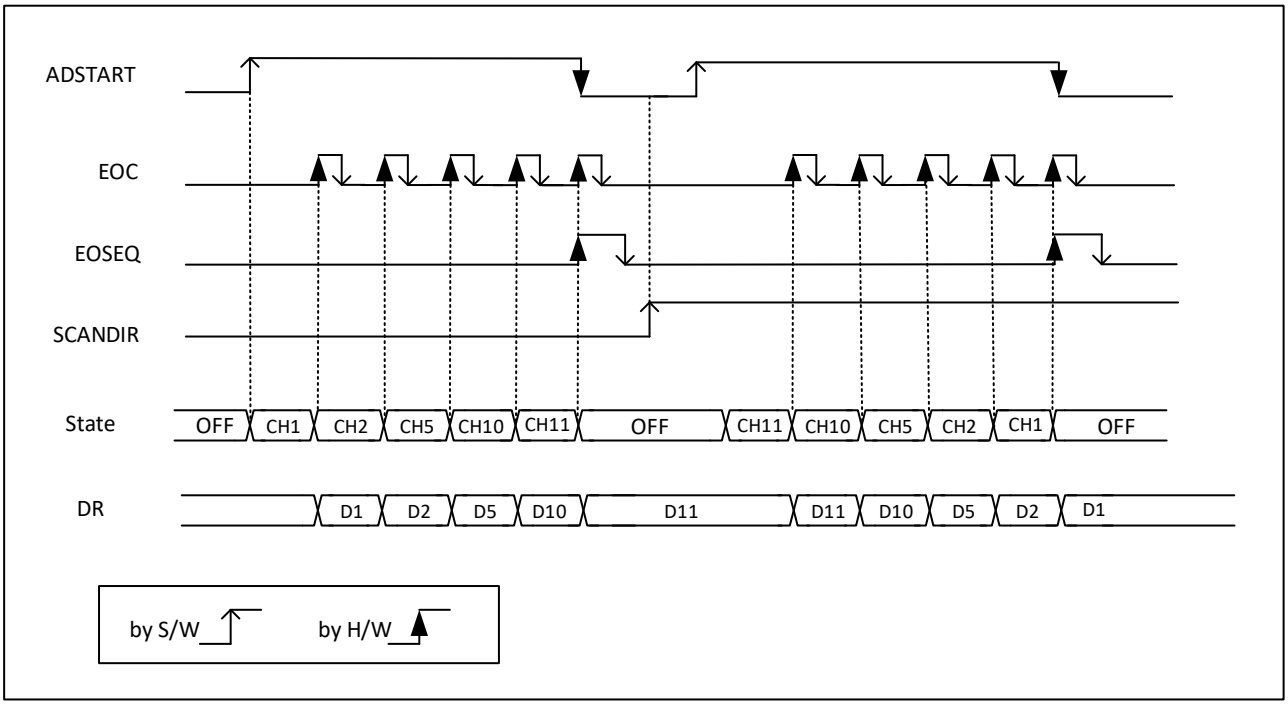


Figure 14-6 Single conversions of a sequence, software trigger

- EXTEN = 0x0, CONT = 0
- CHSEL = 0x20601, WAIT = 0, AUTOFF = 0

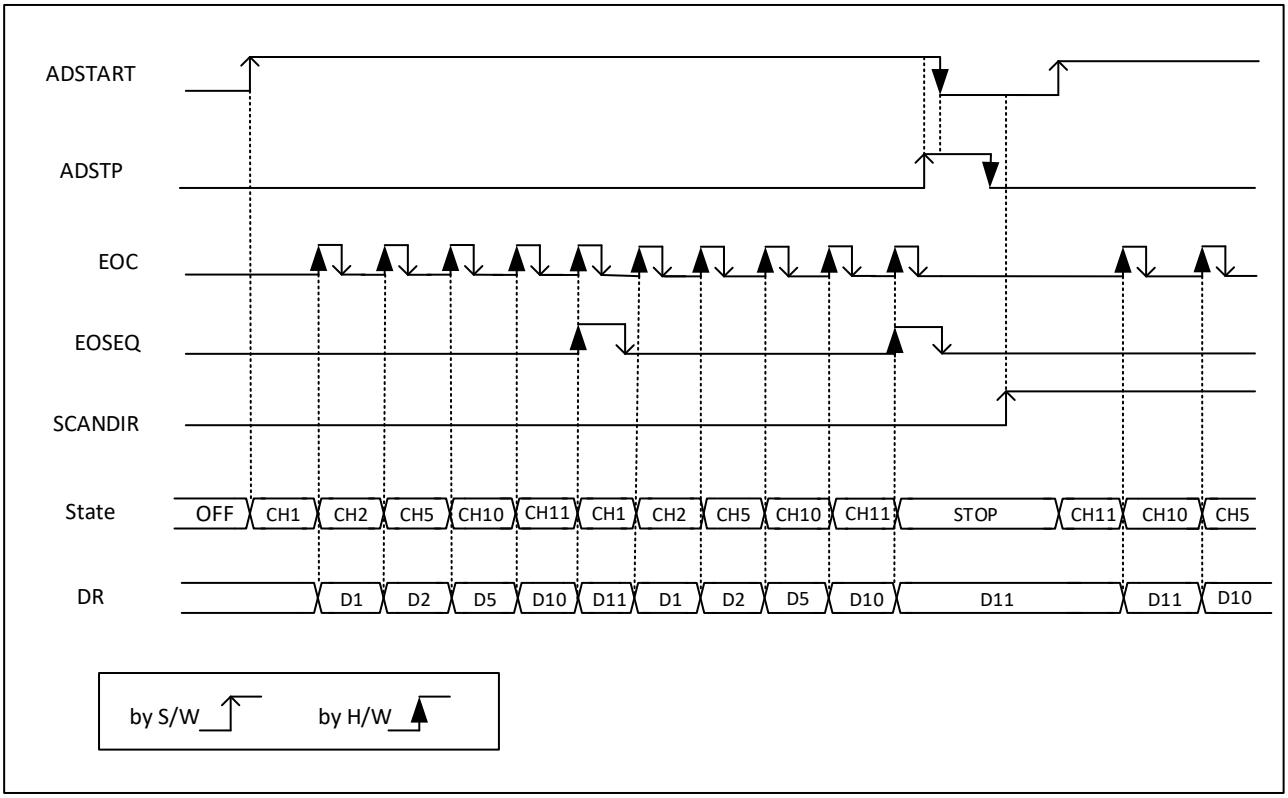


Figure 14-7 Continuous conversion of a sequence, software trigger

- EXTEN = 0x0, CONT = 1
- CHSEL = 0x20601, WAIT = 0, AUTOFF = 0

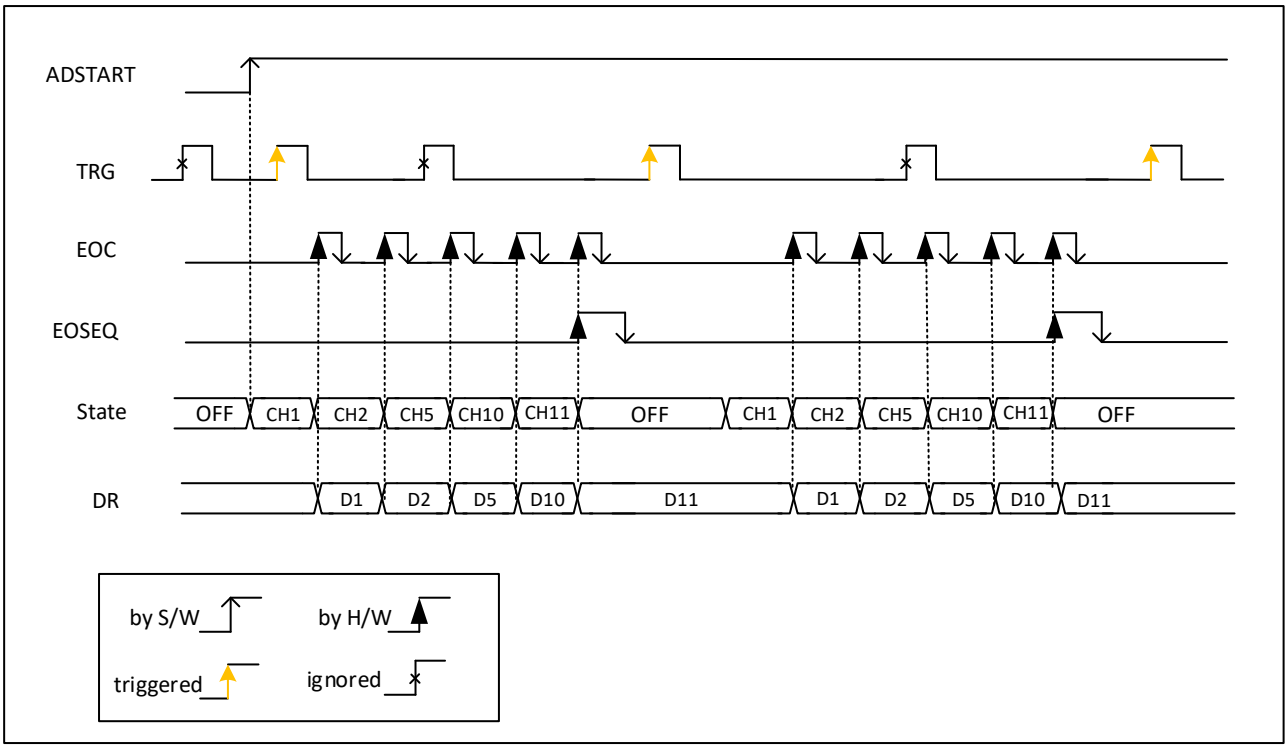


Figure 14-8 Single conversions of a sequence, hardware trigger

- EXTSEL = TRGx, EXTEN = 0x1 (rising edge), CONT = 0
- CHSEL = 0xF, SCANDIR = 0, AUTDLY = 0, AUTOFF = 0

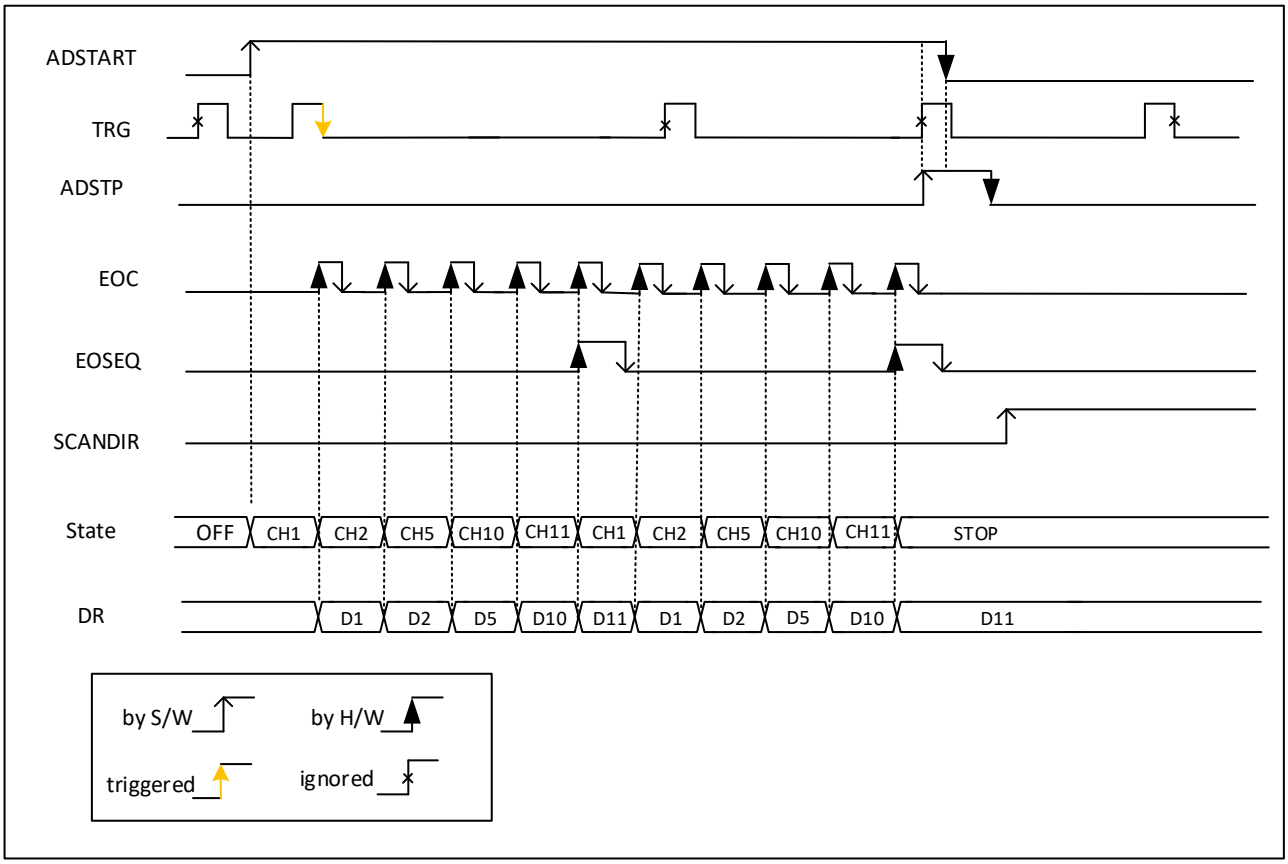


Figure 14-9 Continuous conversion of a sequence, software trigger

- EXTSEL = TRGx, EXTEN = 0x2 (falling edge), CONT = 1
- CHSEL = 0xF, SCANDIR = 0, WAIT = 0, AUTOFF = 0

14.5. Data management

14.5.1. Data register and data alignment (ADC_DR, ALIGN)

At the end of each conversion (when an EOC event occurs), the result of the converted data is stored in the ADC_DR data register which is 16-bit wide.

The format of the ADC_DR depends on the configured data alignment and resolution. The ALIGN bit in the ADC_CFGR1 register selects the alignment of the data stored after conversion. Data can be right-aligned (ALIGN = 0) or left-aligned (ALIGN = 1) .

ALIGN	RESSEL	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0X0	0X0				DATA[11:0]											
	0X1	0X0				DATA[9:0]										0X0	
	0X2	0X0				DATA[7:0]								0x0			
	0X3	0X0				DATA[6:0]						0X0					
1	0X0	DATA[11:0]												0X0			
	0X1	DATA[9:0]										0X0		0X0			
	0X2	DATA[7:0]								0x0				0X0			
	0X3	DATA[6:0]						0X0						0X0			

14.5.2. ADC overrun (OVR, OVRMOD)

The overrun flag (OVR) indicates a data overrun event, when the converted data was not read in time by the CPU or the DMA, before the data from a new conversion is available.

The OVR flag is set in the ADC_ISR register if the EOC flag is still at ‘1’ at the time when a new conversion completes. An interrupt can be generated if the OVRIE bit is set in the ADC_IER register.

When an overrun condition occurs, the ADC keeps operating and can continue to convert unless the software decides to stop and reset the sequence by setting the ADSTP bit in the ADC_CR register.

The OVR flag is cleared by software by writing 1 to it.

It is possible to configure if the data is preserved or oveRWritten when an overrun event occurs by programming the OVRMOD bit in the ADC_CFGR1 register:

- OVRMOD = 0

– An overrun event preserves the data register from being oveRWritten: the old data is maintained and the new conversion is discarded. If OVR remains at 1, further conversions can be performed but the resulting data is discarded.

- OVRMOD = 1

– The data register is oveRWritten with the last conversion result and the previous unread data is lost. If OVR remains at 1, further conversions can be performed and the ADC_DR register always contains the data from the latest conversion.

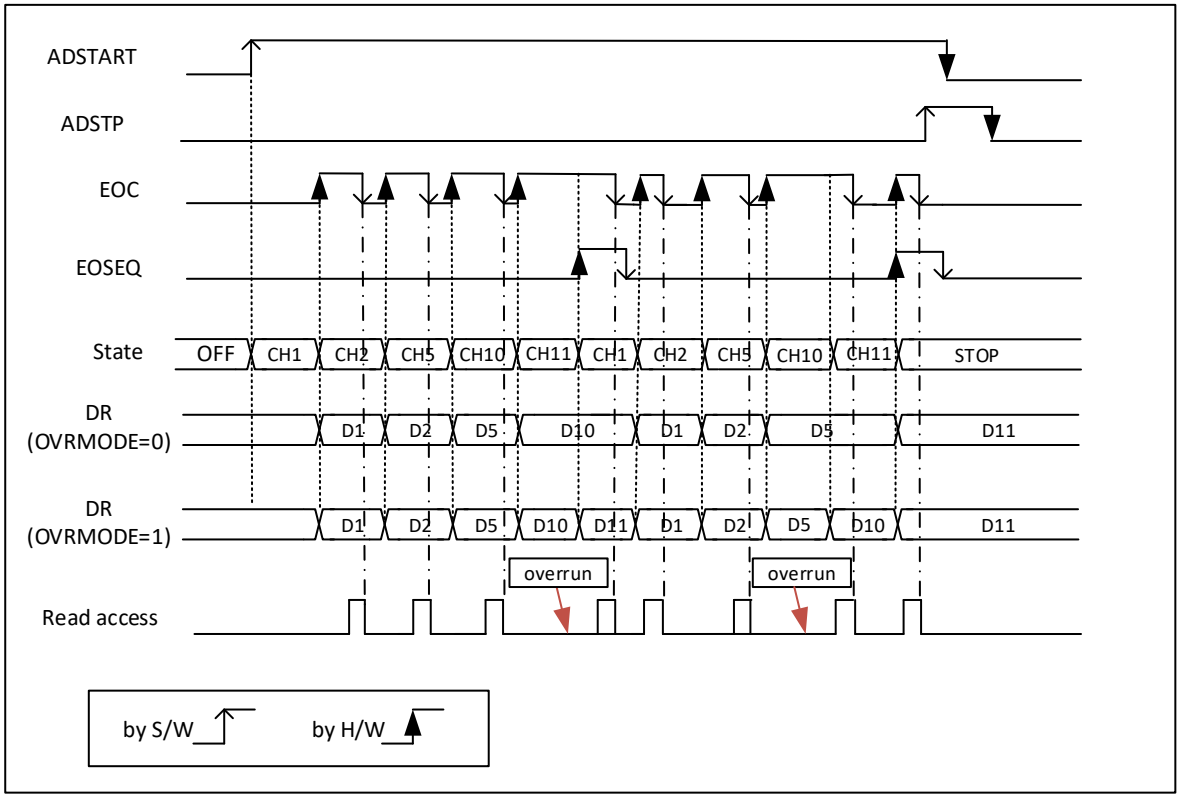


Figure 14-10 Overurn

14.5.3. Managing a sequence of data converted without using the DMA

If the conversions are slow enough, the conversion sequence can be handled by software. In this case the software must use the EOC flag and its associated interrupt to handle each data result. Each time a conversion is complete, the EOC bit is set in the ADC_ISR register and the ADC_DR register can be read. The OVRMOD bit in the ADC_CFGR1 register should be configured to 0 to manage overrun events as an error.

14.5.4. Managing converted data without using the DMA without overrun

It may be useful to let the ADC convert one or more channels without reading the data after each conversion. In this case, the OVRMOD bit must be configured at 1 and the OVR flag should be ignored by the software. When OVRMOD = 1, an overrun event does not prevent the ADC from continuing to convert and the ADC_DR register always contains the latest conversion data.

14.5.5. Managing converted data using the DMA

Since all converted channel values are stored in a single data register, it is efficient to use DMA when converting more than one channel. This avoids losing the conversion data results stored in the ADC_DR register. When DMA mode is enabled (DMAEN bit set to 1 in the ADC_CFGR1 register), a DMA request is generated after the conversion of each channel. This allows the transfer of the converted data from the ADC_DR register to the destination location selected by the software.

Despite this, if an overrun occurs (OVR = 1) because the DMA could not serve the DMA transfer request in time, the ADC stops generating DMA requests and the data corresponding to the new conversion is not transferred by the DMA. Which means that all the data transferred to the RAM can be considered as valid(due to invalid data no longer transmitted).

Depending on the configuration of OVRMOD bit, the data is either preserved or covered.

The DMA transfer requests are blocked until the software clears the OVR bit.

Two different DMA modes are proposed depending on the application use and are configured with bit DMACFG in the ADC_CFGR1 register:

● DMA one shot mode(DMACFG = 0)

This mode should be selected when the DMA is programmed to transfer a fixed number of data words.

● DMA circular mode (DMACFG = 1)

This mode should be selected when programming the DMA in circular mode.

14.5.5.1. DMA one shot mode (DMACFG = 0)

In this mode, the ADC generates a DMA transfer request each time a new conversion data word is available and stops generating DMA requests once the DMA has reached the last DMA transfer even if a conversion has been started again (When the DMA_EOT interrupt is generated, the next ADC conversion may have started).

When the DMA transfer is complete (all the transfers configured in the DMA controller have been done):

● The content of the ADC data register is frozen

● Any ongoing conversion is aborted and its partial result discarded

● No new DMA request is issued to the DMA controller. This avoids generating an overrun error if there are still conversions which are started.

● The scan sequence is stopped and reset

● The DMA is stopped

14.5.5.2. DMA circular mode(DMACFG = 1)

In this mode, the ADC generates a DMA transfer request each time a new conversion data word is available in the data register, even if the DMA has reached the last DMA transfer. This allows the DMA to be configured in circular mode to handle a continuous analog input data stream.

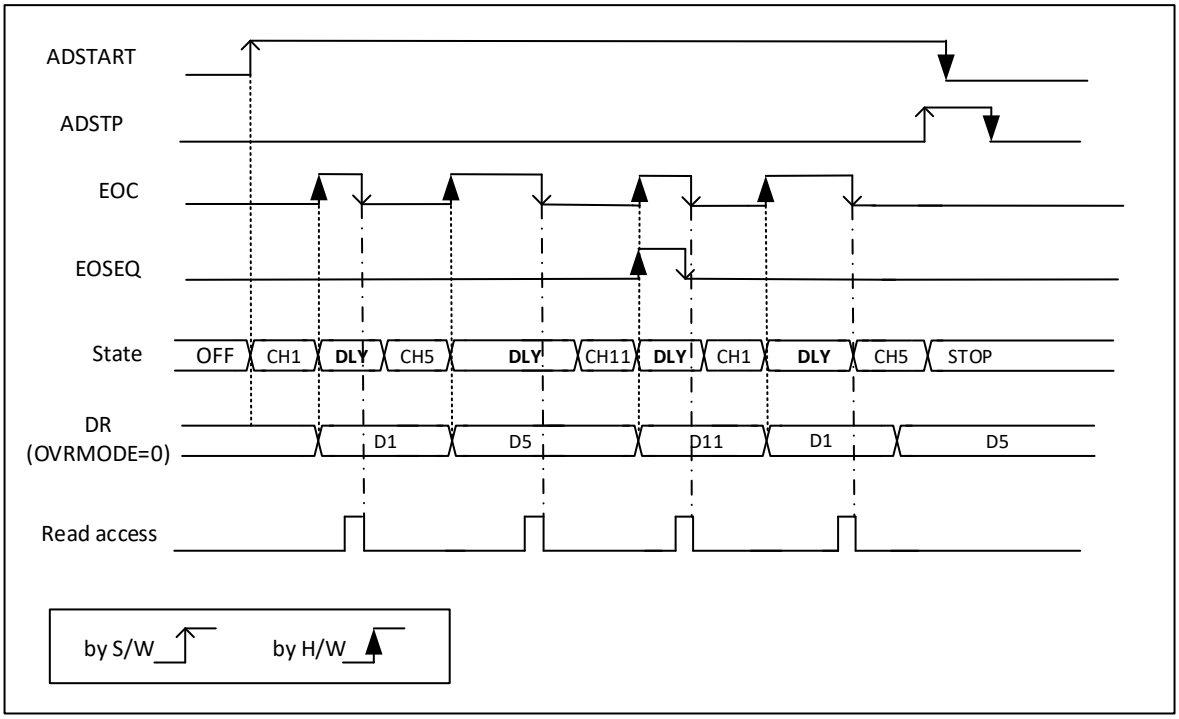
14.6. Low-power features

14.6.1. Wait mode conversion

Wait mode conversion can be used to simplify the software as well as optimizing the performance of applications clocked at low frequency where there might be a risk of ADC overrun occurring.

When the WAIT bit is set to 1 in the ADC_CFGR1 register, a new conversion can start only if the previous data has been treated, once the ADC_DR register has been read or if the EOC bit has been cleared. This is a way to automatically adapt the speed of the ADC to the speed of the system that reads the data.

Note: Any hardware triggers which occur while a conversion is ongoing or during the wait time preceding the read access are ignored.



Figre 14-11 Wait mode conversion

- EXTEN = 0x0, CONT = 1
- CHSEL = 0x3, SCANDIR = 0, AUTDLY = 1, AUTOFF = 0

14.7. Analog window watchdog

The AWD analog watchdog feature is enabled by setting the AWDEN bit in the ADC_CFGR1 register. It is used to monitor that either one selected channel or all enabled channels remain within a configured voltage range (window).

The AWD analog watchdog status bit is set if the analog voltage converted by the ADC is below a lower threshold or above a higher threshold. These thresholds are programmed in the 12 least significant bits of the ADC_HTR and ADC_LTR 16-bit registers. An interrupt can be enabled by setting the AWDIE bit in the ADC_IER register. The AWD flag is cleared by software by writing 1 to it. When converting a data with a resolution of less than 12-bit (according to bits DRES[1:0]), the LSB of the programmed thresholds must be kept cleared because the internal comparison is always performed on the full 12-bit raw converted data (left aligned).

Table 14-3 Analog watchdog comparison

Resolution bits	Analog Watchdog comparison between:	Comments

RES[1:0]	Raw converted data, left aligned	Thresholds	
00: 12-bit	DATA[11:0]	LT[11:0] and HT[11:0]	-
01: 10-bit	DATA[11:2],00	LT[11:0] and HT[11:0]	The user must configure LT[1:0] and HT[1:0] to 00
10: 8-bit	DATA[11:4],0000	LT[11:0] and HT[11:0]	The user must configure LT[3:0] and HT[3:0] to 0000
11: 6-bit	DATA[11:6],000000	LT[11:0] and HT[11:0]	The user must configure LT[5:0] and HT[5:0] to 00000

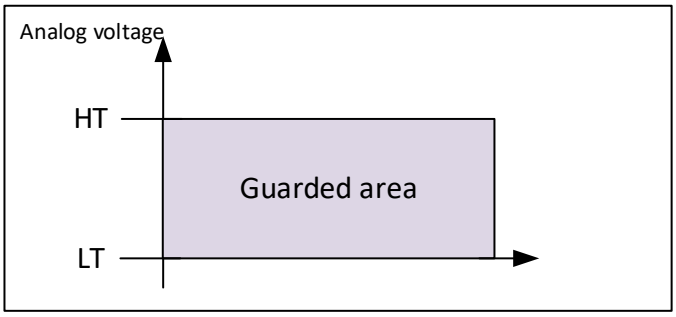


Figure 14-12 Analog watchdog guarded area

Table 14-4 Analog watchdog channel selection

Channels guarded by the analog watchdog	AWDSGL bit	AWDEN bit
None	x	0
All channels	0	1
Single channel	1	1

14.7.1. ADC_AWD_OUT signal output generation

The analog watchdog is associated with an internal hardware signal, ADC_AWD_OUT is directly connected to the ETR input (external trigger) of the on-chip timer TIM1.

When the analog watchdog is enabled, ADC_AWD_OUT is activated:

- When the conversion of the channel selected by AWDCH exceeds the programmed threshold, ADC_AWD_OUT will be set.
- After the conversion of the next channel selected by AWDCH, ADC_AWD_OUT is reset within the programmed threshold. It will remain at 1 if the next protected transition still exceeds the programmed threshold.
- ADC_AWD_OUT is also reset when ADC is disabled. Note that stopping conversion (ADSTP set to 1) may clear the ADC_AWDx_OUT state.
- Channels not selected as analog watchdog do not affect ADC_AWD_OUT status bits.

The AWD flag is set by hardware and reset by software: the AWD flag has no effect on the generation of ADC_AWD_OUT (eg, if the flag is not cleared by software, ADC_AWDx_OUT can toggle while the AWDx flag remains at 1).

The ADC_AWD_OUT signal is generated by the PCLK domain.

AWD comparison is performed at the end of each ADC conversion.

14.8. Temperature sensor and internal reference voltage

A temperature sensor can be used to measure the junction temperature (TJ) of the device.

The temperature sensor is internally connected to the ADC input channel, which can be used to convert the sensor's voltage value to a numerical value. The sampling time of the temperature sensor must be greater than the minimum value of Ts_temp given in the datasheet. When the temperature sensor is not in use, the sensor can be placed in a power-down mode.

The output voltage of the temperature sensor varies linearly with temperature, but each chip has subtle differences related to process variables. In order to improve this accuracy, the calibration value of each chip will be individually given by the product test and saved in the system storage area.

The internal voltage reference (VREFINT) provides a regulated voltage output to the ADC and comparator.

Note: The TSVREF bit must be set to activate two internal channels: temperature sensor, VREFINT.

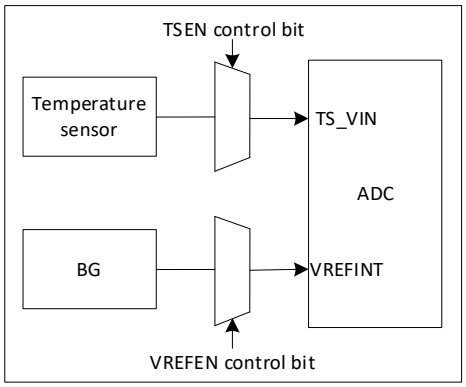


Figure 14-13 TS and VREFINT channel

How to use the temperature sensor to read the temperature:

1. Select ADC1_IN11 input channel
2. Select an appropriate sampling time according to the device specification
3. Set the TSEN bit in the ADC_CCR register to wake up the temperature sensor from power down mode
4. Start ADC conversion with ADSTART bit set in ADC_CR register (external trigger is also available)
5. Read VSENSE conversion data from ADC_DR register
6. Count the temperature using the following formula:

$$Temperature(in\ ℃) = \frac{85℃ - 30℃}{TS_{CAL2} - TS_{CAL1}} \times (TS_{DATA} - TS_{CAL1}) + 30℃$$

TSCAL2 represents the calibration value of the 85°C temperature sensor, the calibration value storage Address offset: 0x1FFF 0F18

TSCAL1 represents the calibration value of the 30°C temperature sensor, the calibration value storage Address offset: 0x1FFF 0F14

TSDATA is the actual output value converted by the ADC

Note: When the sensor wakes up from power-down mode, it needs a start-up time to correctly output VSENSE, and the ADC also has a start-up time after power-on. To reduce this delay, you need to set the ADEN and TSEN bits at the same time.

Calculating the actual Vcc voltage using the internal reference voltage

$$VREFINT = 1.2V = \frac{ADC\_DATAx}{4095} \times VCC$$

Calculating the Vchannnel voltage using the the actual Vcc

$$VCHANNEL = \frac{ADC\_DATAx}{4095} \times VCC$$

VREFINT is fixed at 1.2V,

VCHANNEL is the channel voltage,

ADC_DATA is the conversion data in ADC_DR,

4096 is represented as 12 bits.

14.9. ADC interrputs

ADC interrupts can be generated by any of the following events:

● End of any conversion (EOC flag)

● End of sequence conversion (EOS flag)

● When analog watchdog detection occurs (AWD flag)

● Occurs when the sampling phase ends (EOSMP flag)

● When data overshoot occurs (OVR flag)

Separate interrupt enable bit for flexible setting of ADC interrupts

Table 14-5 ADC interrput

Interrupt event	Event flag	Enable control bit
End of conversion	EOC	EOCIE
End of sequence of conversions	EOS	EOSIE
Analog watchdog status bit is set	AWD	AWDIE
End of sampling phase	EOSMP	EOSMPIE
Overrun	OVR	OVRIE

14.10. ADC regiseters

14.10.1. ADC interrupt and status register (ADC_ISR)

Address offset: 0x00

Reset value: 0x0000 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	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	Res	AWD	Res	Res	OVR	EOSEQ	EOC	EOSMP	Res
								rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
7	AWD	RC_W1	0	Analog watchdog flag This bit is set by hardware when the converted voltage crosses the values programmed in the ADC_LTR and ADC_HTR registers. It is cleared by software writing 1 to it. 0: No analog watchdog event occurred (or the flag event was already acknowledged and cleared by software) 1: Analog watchdog event occurred
6:5	Reserved			
4	OVR	RC_W1	0	ADC overrun

				This bit is set by hardware when an overrun occurs, meaning that a new conversion has complete while the EOC flag was already set. It is cleared by software writing 1 to it. 0: No overrun occurred (or the flag event was already acknowledged and cleared by software) 1: Overrun has occurred
3	EOSEQ	RC_W1	0	End of sequence flag This bit is set by hardware at the end of the conversion of a sequence of channels selected by the CHSEL bits. It is cleared by software writing 1 to it. 0: Conversion sequence not complete (or the flag event was already acknowledged and cleared by software) 1: Conversion sequence complete
2	EOC	RC_W1	0	End of conversion flag This bit is set by hardware at the end of each conversion of a channel when a new data result is available in the ADC_DR register. It is cleared by software writing 1 to it or by reading the ADC_DR register. 0: Channel conversion not complete (or the flag event was already acknowledged and cleared by software) 1: Channel conversion complete
1	EOSMP	RC_W1	0	End of sampling flag This bit is set by hardware during the conversion, at the end of the sampling phase.It is cleared by software by programming it to ‘1’. 0: Not at the end of the sampling phase (or the flag event was already acknowledged and cleared by software) 1: End of sampling phase reached
0	Reserved			

14.10.2. ADC interrupt enable register (ADC_IER)

Address offset: 0x04

Reset value: 0x0000 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	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	Re. s	AWDIE	Res	Res	OVRIE	EOSEQIE	EOCIE	EOSMPIE	Res
								RW			RW	RW	RW	RW	

Bit	Name	R/W	Reset Value	Function
31:8	Reserved			
7	AWDIE	RW	0	Analog watchdog interrupt enable This bit is set and cleared by software to enable/disable the analog watchdog interrupt. 0: Analog watchdog interrupt disabled 1: Analog watchdog interrupt enabled
6:5	Reserved			
4	OVRIE	RW	0	Overrun interrupt enable This bit is set and cleared by software to enable/disable the overrun interrupt. 0: Overrun interrupt disabled 1: Overrun interrupt enabled. An interrupt is generated when the OVR bit is set.
3	EOSEQIE	RW	0	End of conversion sequence interrupt enable This bit is set and cleared by software to enable/disable the end of sequence of conversions interrupt. 0: EOSEQ interrupt disabled 1: EOSEQ interrupt enabled. An interrupt is generated when the EOSEQ bit is set.
2	EOCIE	RW	0	End of conversion interrupt enable

				This bit is set and cleared by software to enable/disable the end of conversion interrupt. 0: EOC interrupt disabled 1: EOC interrupt enabled. An interrupt is generated when the EOC bit is set.
1	EOSMPIE	RW	0	End of sampling flag interrupt enable This bit is set and cleared by software to enable/disable the end of the sampling phase interrupt. 0: EOSMP interrupt disabled. 1: EOSMP interrupt enabled. An interrupt is generated when the EOSMP bit is set.
0	Reserved			

Description: Software can write these bits when ADSTART = 0 (to ensure that no conversion is in progress)

14.10.3. ADC control register (ADC_CR)

Address offset: 0x08

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	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	Res	Res	Res	Res	ADSTP	Res	ADSTART	Res	ADEN
											RS		RS		RS

Bit	Name	R/W	Reset Value	Function
31	ADCAL	RS	0	ADC calibration This bit is set by software to start the calibration of the ADC. It is cleared by hardware after calibration is complete. 0: Calibration complete 1: Write 1 to calibrate the ADC. Read at 1 means that a calibration is in progress.
30:5	Reserved			
4	ADSTP	RS	0	ADC stop conversion command This bit is set by software to stop and discard an ongoing conversion (ADSTP Command). It is cleared by hardware when the conversion is effectively discarded and the ADC is ready to accept a new start conversion command. 0: No ADC stop conversion command ongoing 1: Write 1 to stop the ADC. Read 1 means that an ADSTP command is in progress.
3	Reserved			
2	ADSTART	RS	0	ADC start conversion command This bit is set by software to start ADC conversion. Depending on the EXTEN [1:0] configuration bits, a conversion either starts immediately (software trigger configuration) or once a hardware trigger event occurs (hardware trigger configuration). It is cleared by hardware: – In single conversion mode (CONT = 0, DISCEN = 0), when software trigger is selected

				(EXTEN = 00): at the assertion of the end of Conversion Sequence (EOSEQ) flag. – In discontinuous conversion mode(CONT = 0, DISCEN = 1), when the software trigger is selected (EXTEN = 00): at the assertion of the end of Conversion (EOC) flag. – In all other cases: after the execution of the ADSTP command, at the same time as the ADSTP bit is cleared by hardware. Note: Software is allowed to set ADSTART only when ADEN = 1 (ADC is enabled and there is no pending request to disable the ADC)
1	Reserved			
0	ADEN	RS	0	ADC enable command Software setting this bit enables the ADC and the ADC will be ready to operate. 0: ADC disabled (OFF state) 1: enable ADC

14.10.4. ADC configuration register 1 (ADC_CFGR1)

Address offset: 0x0C

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	AWDCH				Res	Res	AWDEN	AWDSGL	Res	Res	Res.	Res	Res	DISCEN
		RW	RW	RW	RW			RW	RW						RW
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	WAIT	CONT	OVRMOD	Res	Res	Res	EXTSEL			ALIGN	RES_SEL		SCANIR	DMAFG	DMAEN
	RW	RW	RW				RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:30	Reserved			
29:26	AWDCH[3:0]	RW	0000	Analog watchdog channel selection, software can clear and set this bit. Analog Watchdog Monitors Selected Input Channels 0000: ADC analog input channel 0 0001: ADC analog input channel 1 0010: ADC analog input channel 2 …. 1011: Reserved 1011: ADC analog input channel 11 1100: ADC analog input channel 12 Other values: reserved bits Note: The channel configured by the AWDCH[3:0] bits also needs to be set to the CHSELR register Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
25:24	Reserved			
23	AWDEN	RW	0	Analog Watchdog Enable Software can set and clear this bit 0: Disable analog watchdog 1: Enable watchdog Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
22	AWDSGL	RW	0	Enable analog watchdog on one channel or all channels Software can set and clear this bit to enable the analog watchdog on the channel or all channels set by the AWDCH[3:0] bits 0: Enable analog watchdog on all channels 1: Enable analog watchdog on one channel Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)

21:17	Reserved			
16	DISCEN	RW	0	discontinuous mode Software can set and clear this bit to enable/disable discontinuous mode 0: Disable discontinuous mode 1: Enable discontinuous mode It is not possible to enable both discontinuous and continuous modes, setting DISCEN = 1 and CONT = 1 is prohibited. Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
15	Reserved			
14	WAIT	RW	0	wait for conversion mode Software can set and clear this bit to enable/disable wait for conversion mode 0: wait for conversion mode to close 1: Wait for conversion mode to open Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
13	CONT	RW	0	Single/Continuous Conversion Mode Software can set and clear this bit. If set to 1, the conversion will occur consistently until the bit is cleared It is not possible to enable both discontinuous and continuous modes, setting DISCEN = 1 and CONT = 1 is prohibited. Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
12	OVRMOD	RW	0	Overload Management Mode Software can set and clear this bit to configure how data overload is managed 0: ADC_DR register retains old value when overload occurs 1: When an overload occurs, the ADC_DR register will be oveRWritten by the last conversion result Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
11:10	EXTEN[1:0]	RW	00	External trigger enable and polarity selection Software can set and clear this bit, select drive polarity and enable drive 00: Hardware driver detection disabled (software boot transition) 01: Rising edge hardware drive detection 10: Falling edge hardware driver detection 11: Rising edge and falling edge hardware driver detection Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
9	Reserved			
8:6	EXTSEL[2:0]	RW	000	External trigger selection This bit selects the external event that triggers the start of a conversion 000: TRG0(TIM1_TRG0) 001: TRG1(TIM1_CC4) 010: TRG2(Reserved) 011: TRG3(TIM3_TRGP) 100: TRG4(Reserved) 101: TRG5(Reserved) 110: TRG6(Reserved) 111: TRG7(Reserved)
5	ALIGN	RW	0	Data alignment Software sets and clears this bit to select right or left justification 0: right-aligned 1: Left-aligned Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
4:3	RESSEL[1:0]	RW	00	Data resolution Software sets this bit to select the conversion resolution 00: 12 bits 01: 10 bits 10: 8 bits 11: 6 bits These bits are software operable only when ADEN = 0
2	SCANDIR	RW	0	Scan sequence direction Software can set and clear this bit to select the scan sequence direction 0: Up (from channel 0 to channel 11) 1: Down (from channel 11 to channel 0) Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)

1	DMACFG	RW		DMA Configuration This bit can be set and cleared by software, selects between two DMA modes of operation and is valid when DMAEN = 1 0: DMA single mode selection 1: DMA cycle mode selection Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
0	DMAEN	RW	0	Direct Memory Access Enable Software can set and clear this bit to enable the generation of DMA requests. Manage auto-conversion data with DMA controller 0: DMA is not enabled 1: Enable DMA

14.10.5. ADC configuration register 2 (ADC_CFGR2)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CKMODE				Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
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	Res	Res	Res	Res	Res	Res	Res

Bit	Name	R/W	Reset Value	Function
31:28	CKMODE [3:0]	RW	0	ADC clock mode, software can set and clear this bit to define the clock source of the analog ADC 0000: PCLK 0001: PCLK/2 0010: PCLK/4 0011: PCLK/8 0100: PCLK/16 0101: PCLK/32 0110: PCLK/64 1000: HIS 1001: HSI/2 1010: HSI/4 1011: HSI/8 1100: HSI/16 1101: HSI/32 1110: HSI/64 Note: ADCAL = 0, ADSTART = 0, ADSTP = 0 and ADEN = 0 only when ADC is not enabled). Software is allowed to manipulate these bits
27:0	Reserved			

14.10.6. ADC sampling time register (ADC_SMPR)

Address offset: 0x14

Reset value: 0x0000 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	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	Res	Res	Res	Res	Res	Res	SMP		
													RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:3	Reserved			

2:0	SMP[2:0]	RW	000	Sampling time selection Software configurable bit selects the sampling time for all channels 000: 3.5 ADC clock cycles 001: 5.5 ADC clock cycles 010: 7.5 ADC clock cycles 011: 13.5 ADC clock cycles 100: 28.5 ADC clock cycles 101: 41.5 ADC clock cycles 110: 71.5 ADC clock cycles 111: 239.5 ADC clock cycles Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)

14.10.7. ADC watchdog threshold register (ADC_TR)

Address offset: 0x20

Reset value: 0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	HT											
				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**
Res	Res	Res	Res	LT											
				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:28	Re-served			
27:16	HT[11:0]	RW	0xFFF	Analog Watchdog High Threshold Software configurable to define analog watchdog high threshold Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
15:12	Re-served			
11:0	LT[11:0]	RW	0x000	Analog Watchdog Low Threshold Software configurable to define analog watchdog low threshold Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)

14.10.8. ADC channel selection register (ADC_CHSELR)

Address offset: 0x28

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Res	Res	Res	CHSEL 12	CHSEL 11	Res	CHSEL 9	CHSEL 8	CHSEL 7	CHSEL 6	CHSEL 5	CHSEL 4	CHSEL 3	CHSEL 2	CHSEL 1	CHSEL 0
			RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

Bit	Name	R/W	Reset Value	Function
31:14	Reserved		0	
13	Reserved	RW	0	
12	CHSEL12	RW	0	Channel 12 (VREFINT) select enable 0: Channel12 is not selected for conversion 1: Channel12 is selected for conversion Software is allowed to write this bit only if ADSART = 0 (to ensure no conversions are in progress)
11	CHSEL11	RW	0	Channel 11 (TS) select enable

				0: Channel11 is not selected for conversion 1: Channel11 is selected for conversion Software is allowed to write this bit only if ADSART = 0 (to ensure no conversions are in progress)
10	Reserved	RW	0	
9:0	CHSELx	RW	0x0000	Channel selection These bits are software configurable to define the sequence conversion channel 0: Input channel-x is not selected for conversion 1: Input channel-x is selected for conversion Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)

14.10.9. ADC data register (ADC_DR)

Address offset: 0x40

Reset value: 0x0000 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	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	Name	R/W	Reset Value	Function
31:16	Reserved		0	
15:0	Reserved	R	0x0	Converted data This bit is read-only. The conversion result of the last converted channel is placed in this register. Data is left-aligned or right-aligned.

14.10.10. ADC calibration configuration and status registers (ADC_CCSR)

Address offset: 0x44

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALON.	CALFAIL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
R	RC_W1														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	CALSMP[2:0]		CALSEL	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
		RW		RW											

Bit	Name	R/W	Reset Value	Function
31	CALON	R	0	Calibration flag, indicating that ADC calibration is in progress. 1: ADC calibration in progress 0: ADC calibration has ended or ADC calibration has not been started
30	CALFAIL	RC_W1	0	Calibration fail flag, which shows whether the ADC calibration is successful, used in conjunction with CALON. CALON = 0, CALFAIL = 1: ADC calibration failed CALON = 0, CALFAIL = 0: ADC calibration is successful CALON = 1, CALFAIL = 0: Calibrating CALON = 1, CALFAIL = 1: Invalid state Set by hardware, cleared by software writing 1 or cleared by software writing ADCAL = 1.
29:14	Reserved	-	0	-
13:12	CALSMP[2:0]	RW	0	

				Calibration sample time seletion Configure the number of clock cycles for the sampling phase of calibration based on the following information: 00: 2 ADC clock cycles 01: 4 ADC clock cycles 10: 8 ADC clock cycles 11: 1 ADC clock cycle The longer the cycle of configuring SMP during calibration, the more accurate the calibration result, but this configuration will bring the problem of prolonged calibration cycle
11	CALSEL	RW	0	Calibration content selection bit, used to select the content that needs to be calibrated 1: Calibrate OFFSET and linearity 0: Only calibrate OFFSET
10:0	Reserved	-	0	-

14.10.11. ADC common configuration register (ADC_CCR)

Address offset: 0x308

Reset value: 0x0000 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	TSEN	VREFEN	Res	Res	Res	Res	Res	Res
								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	Res	Res	Res	Res	Res	Res	Res

Bit	Name	R/W	Reset Value	Function
31:24	Reserved			
23	TSEN	RW	0	Temperature sensor enable bit, software can set and clear this bit, enable/disable temperature sensor 0: Disable 1: enable Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
22	VREFEN	RW	0	Reference Vrefint enable bit, software can set and clear this bit, enable/disable reference Vrefint 0: Disable 1: enable Software is allowed to write these bits only when ADSART = 0 (to ensure no conversions are in progress)
21:0	Reserved			

14.10.12. ADC register map

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	ADC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD	Res.	Res.	OVR	EOSEQ	EOC	EOSMP	Res.
	Reset value																									0			0	0	0	0	
0x004	ADC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDIE	Res.	Res.	OVRIE	EOSE-	EOCIE	EOSMPI	Res.
	Reset value																									0			0	0	0	0	
0x	ADC_CR	ADCAL	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.	ADSTP	Res.	AD-	Res.	ADEN

0x08	Reset value	0																														0		0		0					
0x0C	ADC_CFGR1	Res.	Res.	AWDCH[3:0]				Res.	Res.	AWDEN	AWDSGL	Res.	Res.	Res.	Res.	Res.	DISCEN	Res.	WAIT	CONT	OVRMOD	Res.	Res.	Res.	EXTSEL[2:0]			ALIGN	RES_SEL[1:0]		SCANIR	DMAFG	DMAEN								
	Reset value			0	0	0	0			0	0						0		0	0	0				0	0	0	0	0	0	0	0	0							
0x10	ADC_CFGR2	CKMODE[3:0]				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.							
	Reset value	0	0	0	0																																			
0x14	ADC_SMPR	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.	SMP[2:0]									
	Reset value																													0										
0x20	ADC_TR	Res.	Res.	Res.	Res.	HT[11:0]												Res.	Res.	Res.	Res.	LT[11:0]																			
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0	0							
0x28	ADC_CHSELR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHSEL12	CHSEL11	Res.	CHSEL9	CHSEL8	CHSEL7	CHSEL6	CHSEL5	CHSEL4	CHSEL3	CHSEL2	CHSEL1	CHSEL0							
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x40	ADC_DR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA[15:0]																							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							
0x44	ADC_CCSR	CALON	CALFAIL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALSMP[1:0]		CALSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
	Reset value	0	0																	0	0	0																		
0x308	ADC_CCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSEN	VREFE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							
	Reset value									0	0																													

15. Comparator (COMP)

15.1. Introduction

Two general purpose comparators (general purpose comparators) COMP are integrated in the chip, namely COMP1 and COMP2. These two modules can be used as separate modules or combined with timer.

Comparators can be used as follows:

- Triggered by analog signal to generate low power mode wake-up function
- Analog signal conditioning
- Current control loop of Cycle by cycle when connected with PWM output from timer

15.2. COMP main features

- Each comparator has configurable positive or negative input for flexible voltage selection
  - Multiple I/O pins
  - VCC
  - Output of temperature sensor
  - Internal reference voltage and 3 fractional values (1/4, 1/2, 3/4) provided by voltage divider
- Configurable hysteresis function
- Programmable speed and power consumption
- Output can be connected to I/O or timer input as trigger
  - OCREF_CLR event (cycle by cycle current control)
  - Brake for fast PWM shutdown
- COMP1 and COMP2 can be combined into window COMP
- Each COMP has interrupt generation capability, which is used as wake-up (via EXTI) from low-power modes (sleep and stop modes)

15.3. COMP function description

15.3.1. COMP diagram

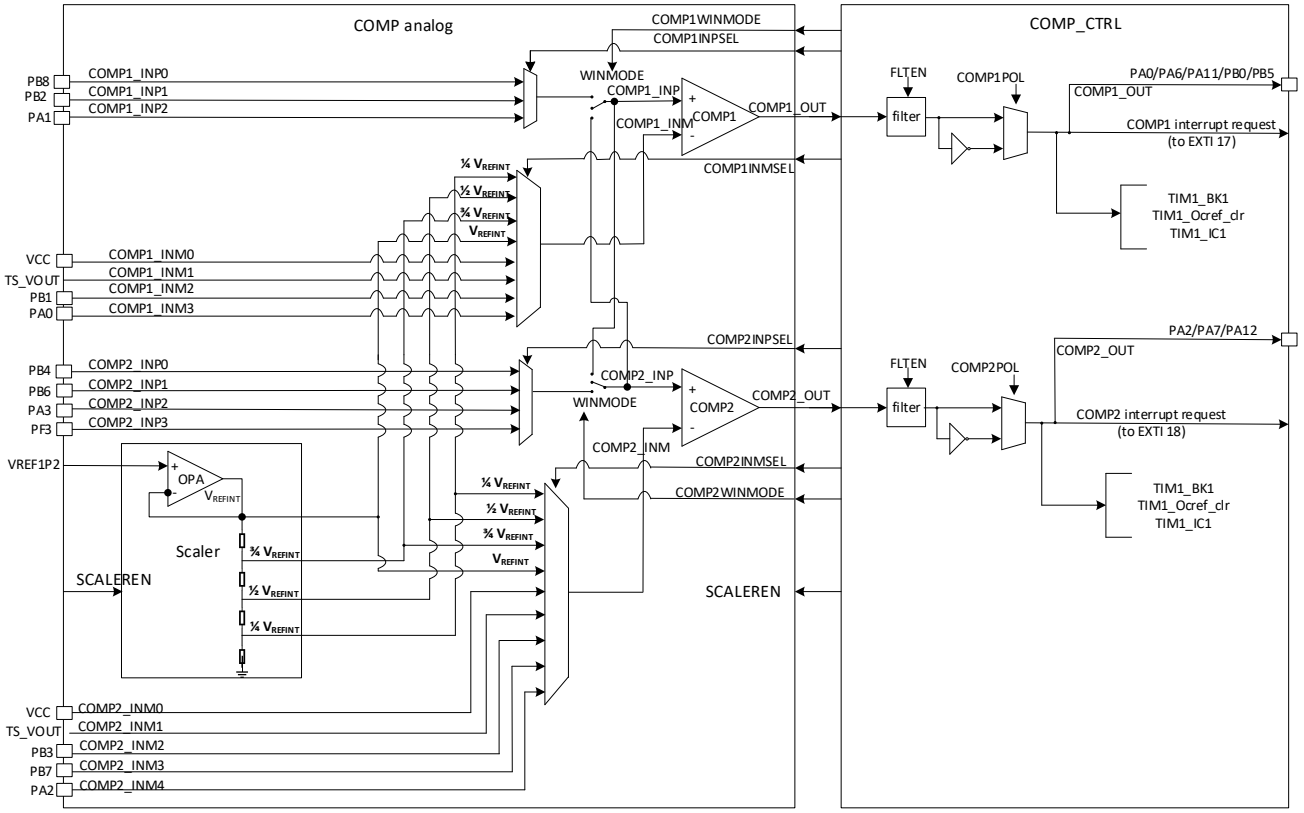


Figure 15-1 Comparator architecture block diagram

15.3.2. COMP pins and internal signals

The I/O used as comparator input must be configured in analog mode in the GPIO register.

The comparator output can be connected to the I/O pin through the alternate function channel (alternate function) on the GPIO.

The outputs can also be internally connected to the inputs of various timers for the following purposes:

- When the brake input is connected, the emergency shutdown of the PWM signal
- Cycle-by-cycle current control using OCREF_CLR input
- Input capture for timing measurements

15.3.3. COMP reset and clock

The COMP module has two clock sources:

1) PCLK (APB clock), used to provide the clock to the configuration register

2) COMP clock, used for the clock of the circuit after the analog comparator output (the latch circuit of the analog output, the glitch filter circuit, etc.), which can be selected as PCLK, LSE or LSI. When you need to work in stop mode, choose LSE or LSI.

The reset signal sources of the COMP module are:

1) The reset of the circuit after the analog comparator output (the latch circuit of the analog output, the glitch filter circuit, etc.), the reset signal includes the APB reset source and the COMP module software reset source (RCC_APBRSTR2.COMP1RST and RCC_APBRSTR2.COMP2RST)

15.3.4. COMP lock mechanism

Comparators can be used for safety purposes such as overcurrent and temperature protection. For applications with specific functional safety requirements, it is necessary to ensure that the comparator program cannot be rewritten in the event of register access errors and PC (program counter) confusion. Thus, the comparator control and status registers can be write protected (read only).

If the write to the register is complete, the COMPx Lock bit is set to 1, which makes the entire register read-only, including the COMPx Lock bit.

Write protection can only be reset by the chip's reset signal.

15.3.5. Window comparator

The role of the Window comparator is to monitor whether the analog voltage is within the low and high thresholds. A window comparator can be created using two comparators. The monitored analog voltage is connected to the non-inverting (+) inputs of both comparators at the same time, and the high and low thresholds are connected to the inverting inputs (-) of the two comparators, respectively.

By enabling the WINMODE bit, the non-inverting (+ input) of the two comparators can be connected together to save one I/O pin.

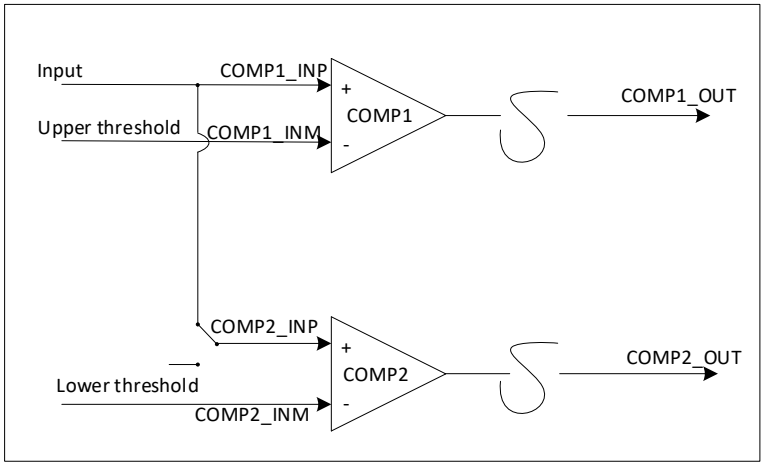


Figure 15-2 Window comparator

15.3.6. Hysteresis

To avoid spurious output transitions in noisy signal conditions, the comparator can be enabled with hysteresis (by enabling the HYST bit in COMP1_CSR, both COMP1 and COMP2 hysteresis can be turned on).

15.3.7. Power modes

The power consumption and propagation delay of the comparator can be selected from different modes by the PWRMODE[1:0] bits of the COMPx_CSR register to achieve the most suitable trade-off in a specific application. The optional modes include high speed and medium speed. Relatively speaking, the high speed mode consumes more power and has a smaller transmission delay. Note that before entering stop, if you select the PWR_CR2 register LPR = 1 (that is, choose to use the low power regulator to supply power), you need to first set COMP at Medium speed (PWRMODE = 01).

In addition, in order to reduce power consumption, APB clock and COMP clock are controlled by RCC_APBENR2.COMP1EN (and RCC_APBENR2.COMP2EN), software can only enable this register when using COMP module.

15.3.8. Comparator filtering

The output filter function of COMP and the corresponding filter width can be enabled by setting the COMP_FR register. Note that this setting should be done before COMP_EN is enabled.

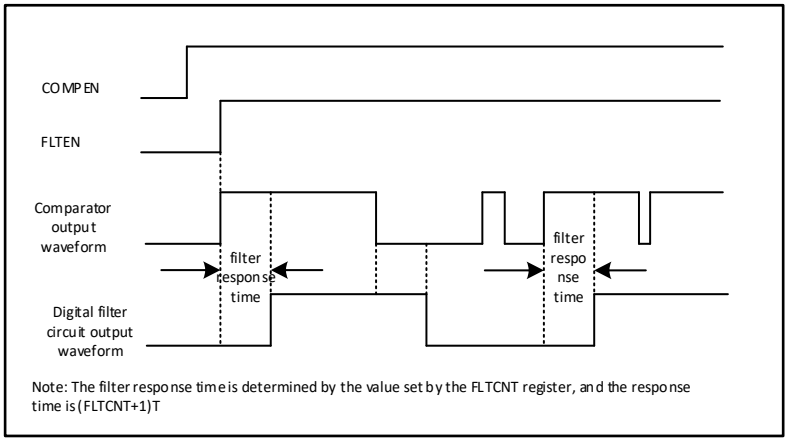


Figure 15-3 COMP filter

15.3.9. COMP interrupt

The comparator output is internally connected to the EXTI controller (extended interrupts and events). Each comparator has a separate EXTI line (17 and 18) and can generate interrupts or events. The same mechanism is used for wake-up from low power.

15.4. COMP registers

15.4.1. COMP1 control and status registers (COMP1_CSR)

Address offset: 0x00

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP_OUT	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PWR-MODE[1:0]		Res	HYST
RW	R											RW	RW		RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO-LAR-ITY	Res	Res	Res	WINMODE	Res	INPSEL[1:0]		INMSEL[3:0]				Res	Res	SCALER_EN	COMP1_EN
RW		-	-	RW	-	RW	RW	RW	RW	RW	RW			RW	RW

Bit	Name	R/W	Reset Value	Function
31	LOCK	RW	0	COMP1_CSR register lock Set by software and cleared by system reset. When set, all 32 bits of the COMP1_CSR register are locked 0: Unlocked, the entire register can be read and written 1: Locked, the entire register is read-only
30	COMP_OUT	R		COMP1 output status This bit is read-only and reflects the polarity-selected output level of COMP1.
29:20	Reserved			
19:18	PWRMODE[1:0]	RW	0	COMP1 power mode selection

				Software is readable and writable, choosing the power consumption and thus the speed of the COMP1. In high speed mode, the power consumption is larger and the delay is smaller 00: High speed 01: Medium speed 10: High speed 11: High speed Note: This bit is not controlled by the LOCK function.
17	Reserved			
16	HYST	RW	0	COMP1 and COMP2 hysteresis function enable control 0: The hysteresis function is disabled 1: The Hysteresis function enabled
15	POLARITY	RW	0	COMP1 polarity selection Software readable and writable (if not locked) 0: do not reverse 1: Reverse
14:12	Reserved			
11	WINMODE	RW	0	COMP1 output selection (window mode) Software readable and writable (if not locked) 0: The signal is selected by INPSEL[1:0] 1: COMP2_INP signal of COMP2 Note that the WINMODE modes of the two COMPs cannot be enabled at the same time.
10	Reserved			
9:8	INPSEL[1:0]	RW	00	00: PB8 01: PB2 10: PA1 11: Reserved
7:4	INMSEL[3:0]	RW	0000	0000: 1/4 VREFINT 0001: 1/2 VREFINT 0010: 3/4 VREFINT 0011: VREFINT 0100: VCC 0101: TS 0110: PB1 0111: Reserved 1000: PA0 其他: 1/4 VREFINT
3:2	Reserved			
1	SCALER_EN	RW	0	The VREFINT related input is enabled. When any one of VREFINT, 3/4 VREFINT, 1/2 VREFINT, and 1/4 VREFINT is selected as the comparator input, this register bit must be turned on. 0: do not open SCALER 1: Enable SCALER
0	COMP1_EN	RW	0	COMP1 enable bit Software readable and writable (if not locked) 0: Disable 1: Enable

15.4.2. COMP1 filter register (COMP1_FR)

Address offset: 0x04

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT1[15:0]															
RW	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**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FLTEN1
															RW

Bit	Name	R/W	Reset Value	Function
31:16	FLTCNT1	RW	0x0	Comparator 1 Sample Filter Counter

				The sampling clock is APB or LSI or LSE. The filter count value is configurable. When the number of sampling times reaches the filter count value, the results are output consistently. Sampling count period = FLTCNT[15:0]
15:1	Reserved		0x0	
0	FLTEN1	RW	0x0	Comparator 1 digital filter function configuration 0: Disable digital filter function 1: Enable digital filter function Note: This bit must be set when COMP1_EN is 0

15.4.3. COMP2 control and status registers (COMP2_CSR)

Address offset: 0x10

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	COMP_OUT	Res	Res	Res	Res	Res	Res					PWR-MODE[1:0]		Res	
RW	R											RW	RW		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PO-LAR-ITY	Res	Res	Res	WINMODE	Res	INPSEL[1:0]		INMSEL[3:0]				Res	Res	Res	COMP2_EN
RW		-	-	RW	-	RW	RW	RW	RW	RW	RW				RW

Bit	Name	R/W	Reset Value	Function
31	LOCK	RW	0	COMP2_CSR register lock Set by software and cleared by system reset. When set, all 32 bits of the COMP2_CSR register are locked 0: Unlocked, the entire register can be read and written 1: Locked, the entire register is read-only
30	COMP_OUT	R		COMP2 output status This bit is read-only and reflects the polarity-selected output level of COMP2.
29:20	Reserved			
19:18	PWRMODE[1:0]	RW		COMP2 power mode selection Software readable and writable, power mode selected and the resulting speed of COMP2 00: High speed 01: Medium speed 10: High speed 11: High speed Note: This bit is not controlled by the LOCK function.
17:16	reserved			
15	POLARITY	RW		COMP2 polarity selection Software readable and writable (if not locked) 0: do not reverse 1: Reverse
14:12	Reserved			
11	WINMODE	RW		COMP2 non-inverting output selection (window mode) Software readable and writable (if not locked) 0: The signal is selected by INPSEL[1:0] 1: COMP2_INP signal of COMP2 Note that the WINMODE modes of the two COMPs cannot be enabled at the same time.
10	Reserved			
9:8	INPSEL[1:0]	RW		Signal selection of COMP2 non-inverting input Software readable and writable (if not locked) 00: PB4 01: PB6 10: PA3 11: PF3
7:4	INMSEL[3:0]	RW		0000: 1/4 VREFINT 0001: 3/4 VREFINT

				0010: 1/2 VREFINT 0011: VREFINT 0100: VCC 0101: TS 0110: PB3 0111: PB7 1000: PA2 > 1000: 1/4 VREFINT
3:1	Reserved			
0	COMP2_EN	RW		COMP2 enable bit Software readable and writable (if not locked) 0: Disable 1: Enable

15.4.4. COMP2 filter register (COMP2_FR)

Address offset: 0x14

Reset value: 0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLTCNT2[15:0]															
RW	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**
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FLTEN2
															RW

Bit	Name	R/W	Reset Value	Function
31:16	FLTCNT2[15:0]	RW	0x0	Comparator 2 Sample Filter Counter The sampling clock is APB or LSI or LSE. The filter count value is configurable. When the number of sampling times reaches the filter count value, the results are output consistently. Sampling count period = FLTCNT[15:0]
15:1	Reserved		0x00	
0	FLTEN2	RW	0x0	Comparator 2 digital filter function configuration 0: Disable digital filter function 1: Enable digital filter function Note: This bit must be set when COMP2_EN is 0

15.4.5. COMP register map

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	COMP1_CSR	LOCK	COMP_OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWR-MODE[1:0]		Res.	HYST	POLARITY	Res.	Res.	Res.	WINMODE	Res.	INPSEL[1:0]		INMSEL [3:0]				Res.	Res.	SCALER_EN	COMP1_EN
	Reset value	0	0											0	0		0	0				0		0	0	0	0	0	0			0	0
0x004	COMP1_FR	FLTCNT1[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																0
0x010	COMP2_CSR	LOCK	COMP_OUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWR-MODE[1:0]		Res.	HYST	POLARITY	Res.	Res.	Res.	WINMODE	Res.	INPSEL[1:0]		INMSEL [3:0]				Res.	Res.	SCALER_EN	COMP2_EN
	Reset	0	0											0	0		0	0				0		0	0	0	0	0	0			0	0

	value																														
0x14	COMP2_FR	FLTCNT2[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLTEN2
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0														0

16. LED Controller

16.1. Introduction

This LED controller supports the control function of 1 to 4 8-segment common cathode LED digital tubes. The controller lights up four 7-segment digital tubes corresponding to the output through four pins (PB) that support super-large sink current (80mA/60mA/40mA/20mA configurable), and only one number is lighted at the same time.

16.2. LED function description

- **Diagram**

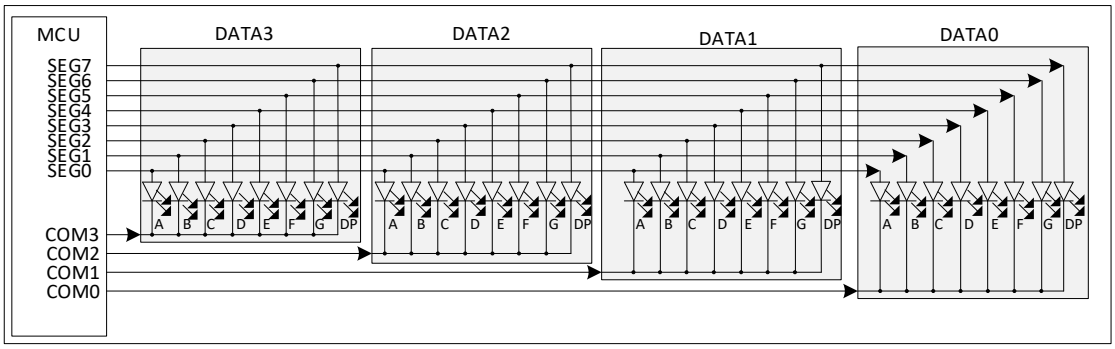


Figure 16-1 System application block diagram

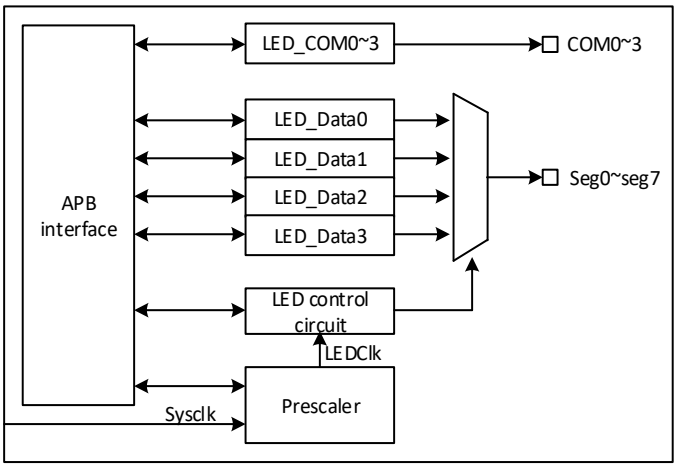


Figure 16-2 LED architecture diagram

- **LED controller sequence**

The LED controller can drive 1-4 digital tubes, and the waveforms are described as follows:

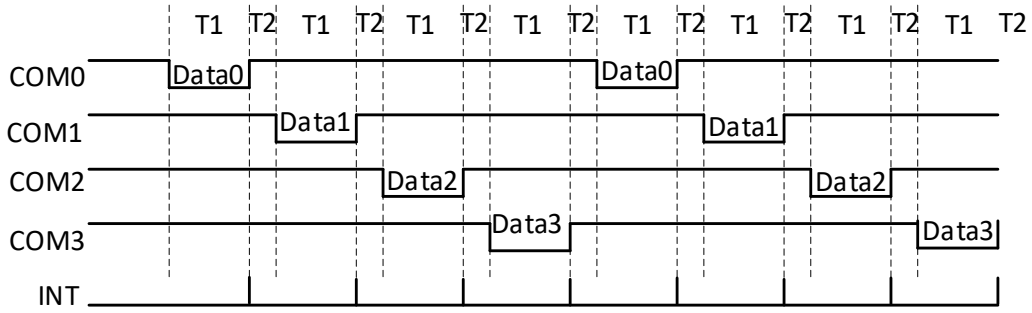


Figure 16-3 LED controller sequence

The software selects the digital tube to be lit by writing the LED_DRx register.

Configure the LED_TR register through software to generate the time of T1 and T2 to achieve the effect of displaying digital scanning. After each T1 (that is, the time for lighting a number), you can choose whether to generate an interrupt. During T2 time, the output of GPIO will be turned off. At the same time, only one number can be lit.

16.3. LED registers

16.3.1. Control register (LED_CR)

Address offset: 0x00

Reset value: 0x0000 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	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	EHS	Res	Res	Res	Res	Res	Res	Res	Res	IE	LED_COM_SEL[1:0]		LED ON
			RW									RW	RW		RW

Bit	Name	R/W	Reset Value	Function
31:13	Reserved		-	
12	EHS	RW	00	LED tube brightness level setting control bit. 0: Non-LED COM output 1: LED COM output, IO is the largest drive capability. Cooperate with GPIO_OSPEEDR of LED IO (PA15, PB3, PB4, PB5) to form the following control of driving capability (brightness): 00: Level 0, all IO pins selected as COM ports have a current sink capacity of 20mA 01: Level 1, all IO pins selected as COM ports have a current sink capacity of 40mA 10: Level 2, all IO pins selected as COM ports have a current sink capacity of 60mA 11: Level 3, all IO pins selected as COM ports have a current sink capacity of 80mA
11:4	Reserved			
3	IE	RW	0	LED Interrupt Enable Bit 0: LED interrupt disabled 1: LED interrupt enable
2:1	LED_COM_SEL[1:0]	RW	00	LED digital tube selection 00: 1 digit 01: 2 numbers light up in turn 10: 3 numbers light up in turn 11: 4 numbers light up in turn
0	LEDON	RW	0	LED enable control bit 0: LED is not enabled

				1: LED enable

16.3.2. Prescaler register (LED_PR)

Address offset: 0x04

Reset value: 0x0000 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	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	Res	PR[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7:0	PR[7:0]	RW	0x00	LED prescale factor control bits. $f_{LED} = f_{PCLK}/(PR+1$ f_{LED} is the clock frequency of timer operation count $f_{PCLK}$is the chip system clock frequency PRis the prescaler factor

16.3.3. Time register (LED_TR)

Address offset: 0x08

Reset value: 0x0000 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	Res	Res	Res
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
T2[7:0]								T1[7:0]							
RW								RW							

Bit	Name	R/W	Reset Value	Function
31:16	Reserved		-	
15:8	T2[7:0]	RW	0x00	LED switching time control, T2 counts under f_{LED} Note: T2 cannot be 0, otheRWise the hardware will not start the scan sequence.
7:0	T1[7:0]	RW	0x00	LED lighting time control, T1 is counted under f_{LED}

16.3.4. Data register 0 (LED_DR0)

Address offset: 0x0C

Reset value: 0x0000 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	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	Res	DATA0[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7	DATA0_DP	RW	0	The DP tube of DATA0 lights up
6	DATA0_G	RW	0	The G tube of DATA0 lights up
5	DATA0_F	RW	0	The F tube of DATA0 lights up
4	DATA0_E	RW	0	The E tube of DATA0 lights up
3	DATA0_D	RW	0	The D tube of DATA0 lights up
2	DATA0_C	RW	0	The C tube of DATA0 lights up
1	DATA0_B	RW	0	The B tube of DATA0 lights up
0	DATA0_A	RW	0	The A tube of DATA0 lights up

16.3.5. Data register 1 (LED_DR1)

Address offset: 0x10

Reset value: 0x0000 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	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	Res	DATA1[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7	DATA1_DP	RW	0	The DP tube of DATA1 lights up
6	DATA1_G	RW	0	The G tube of DATA1 lights up
5	DATA1_F	RW	0	The F tube of DATA1 lights up
4	DATA1_E	RW	0	The E tube of DATA1 lights up
3	DATA1_D	RW	0	The D tube of DATA1 lights up
2	DATA1_C	RW	0	The C tube of DATA1 lights up
1	DATA1_B	RW	0	The B tube of DATA1 lights up
0	DATA1_A	RW	0	The A tube of DATA1 lights up

16.3.6. Data register 2 (LED_DR2)

Address offset: 0x14

Reset value: 0x0000 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	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	Res	DATA2[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7	DATA2_DP	RW	0	The DP tube of DATA2 lights up
6	DATA2_G	RW	0	The G tube of DATA2 lights up
5	DATA2_F	RW	0	The F tube of DATA2 lights up
4	DATA2_E	RW	0	The E tube of DATA2 lights up
3	DATA2_D	RW	0	The D tube of DATA2 lights up
2	DATA2_C	RW	0	The C tube of DATA2 lights up
1	DATA2_B	RW	0	The B tube of DATA2 lights up
0	DATA2_A	RW	0	The A tube of DATA2 lights up

16.3.7. Data register 3 (LED_DR3)

Address offset: 0x18

Reset value: 0x0000 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	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	Res	DATA3[7:0]							
								RW							

Bit	Name	R/W	Reset Value	Function
31:8	Reserved		-	
7	DATA3_DP	RW	0	The DP tube of DATA3 lights up
6	DATA3_G	RW	0	The G tube of DATA3 lights up
5	DATA3_F	RW	0	The F tube of DATA3 lights up
4	DATA3_E	RW	0	The E tube of DATA3 lights up

3	DATA3_D	RW	0	The D tube of DATA3 lights up
2	DATA3_C	RW	0	The C tube of DATA3 lights up
1	DATA3_B	RW	0	The B tube of DATA3 lights up
0	DATA3_A	RW	0	The A tube of DATA3 lights up

16.3.8. Interrupt register (LED_IR)

Address offset: 0x1C

Reset value: 0x0000 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	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	Res	Res	Res	Res	Res	Res	Res	Res	FLAG
															RC_W1

Bit	Name	R/W	Reset Value	Function
31:1	Reserved		-	
0	FLAG	RC_W1	0	Set by hardware and cleared by software writing 1. LED complete interrupt flag bit for single digit lighting 0: LED has not completed the lighting of a single digit 1: LED completes the lighting of a single digit

16.3.9. LED register map

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	LED_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EHS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IE	LED_COMP_SEL[1:0]		LED_ON
	Reset value																				0									0	0	0	0
0x004	LED_PR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x008	LED_TR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	T2[7:0]								T1[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	LED_DR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA0[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x010	LED_DR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATA1[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

0x14	LED_DR2	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.	DATA2[7:0]										
	Reset value																										0	0	0	0	0	0	0	0			
0x18	LED_DR3	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.	DATA3[7:0]										
	Reset value																										0	0	0	0	0	0	0	0			
0x1C	LED_IR	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.	Res.	Res.	Res.	FLAG
	Reset value																																				0