



---

# How to use the cache to optimize performance and power efficiency on STM32 MCUs

## Introduction

This application note describes the instruction cache (ICACHE) and the data cache (DCACHE), the first caches developed by STMicroelectronics.

The ICACHE and DCACHE introduced on the AHB bus of the Arm Cortex<sup>®</sup>-M33 processor are embedded in the STM32 microcontroller (MCUs) listed in the table below. These caches allow users to improve their application performance and reduce the consumption when fetching instruction and data from both internal and external memories, or for data traffic from external memories.

This document gives typical examples to highlight the ICACHE and DCACHE features and facilitate their configuration.

**Table 1. Applicable products**

Type	Product series
Microcontrollers	STM32H5, STM32L5, STM32U3, STM32U5, STM32C5

## 1 General information

---

This application note applies to the STM32 series microcontrollers that are Arm Cortex® core-based devices.

*Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.*

**arm**

## 2 ICACHE and DCACHE overview

This section provides an overview of the ICACHE and DCACHE interfaces embedded in the STM32 Arm Cortex® core-based microcontrollers.

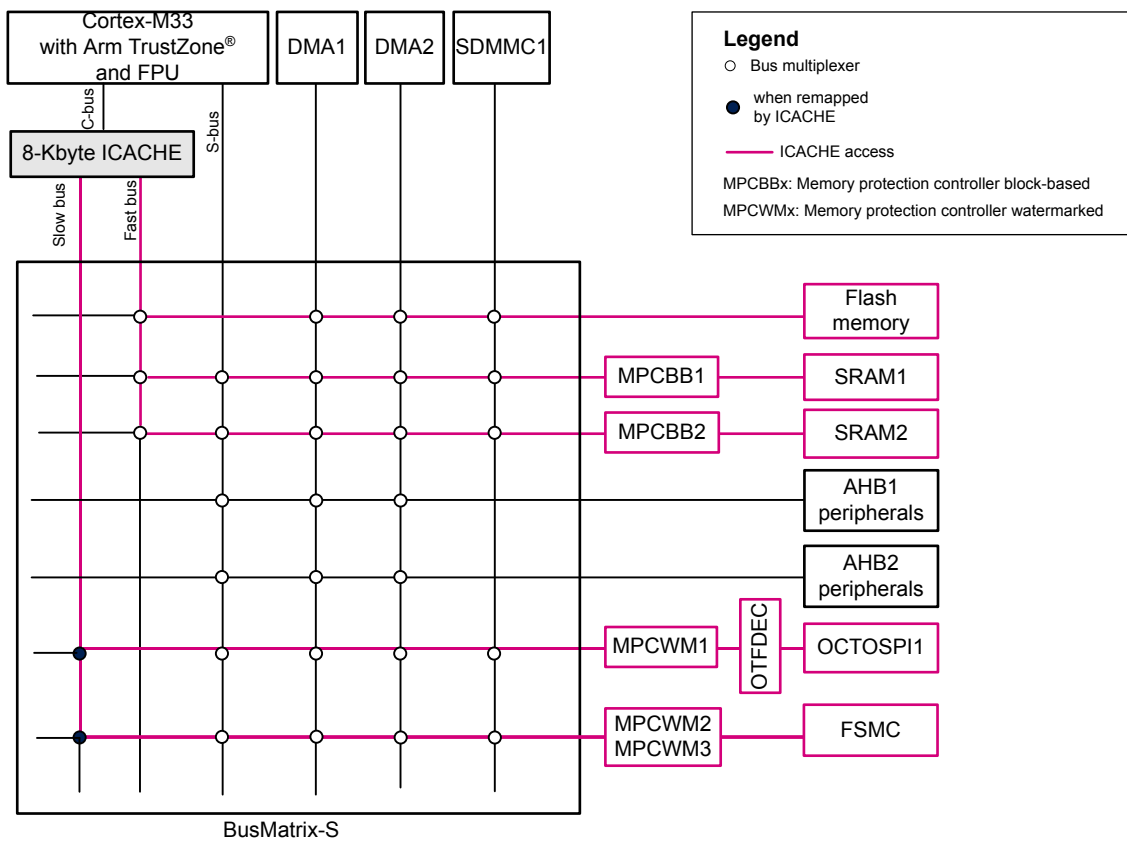
This section details the ICACHE and DCACHE diagram and integration in the system architecture.

### 2.1 STM32L5 series smart architecture

The architecture is based on a bus matrix allowing multiple masters (Cortex®-M33, ICACHE, DMA1/2, and SDMMC1) to access multiple slaves (such as flash memory, SRAM1/2, OCTOSPI1, or FSMC).

The figure below describes the STM32L5 series smart architecture.

Figure 1. STM32L5 series smart architecture



The Cortex®-M33 performance is improved by using the 8-Kbyte ICACHE interface introduced to its C-AHB bus, when fetching code or data from the internal memories (flash memory, SRAM1, or SRAM2) through the fast bus, and also from the external memories (OCTOSPI1 or FSMC) through the slow bus.

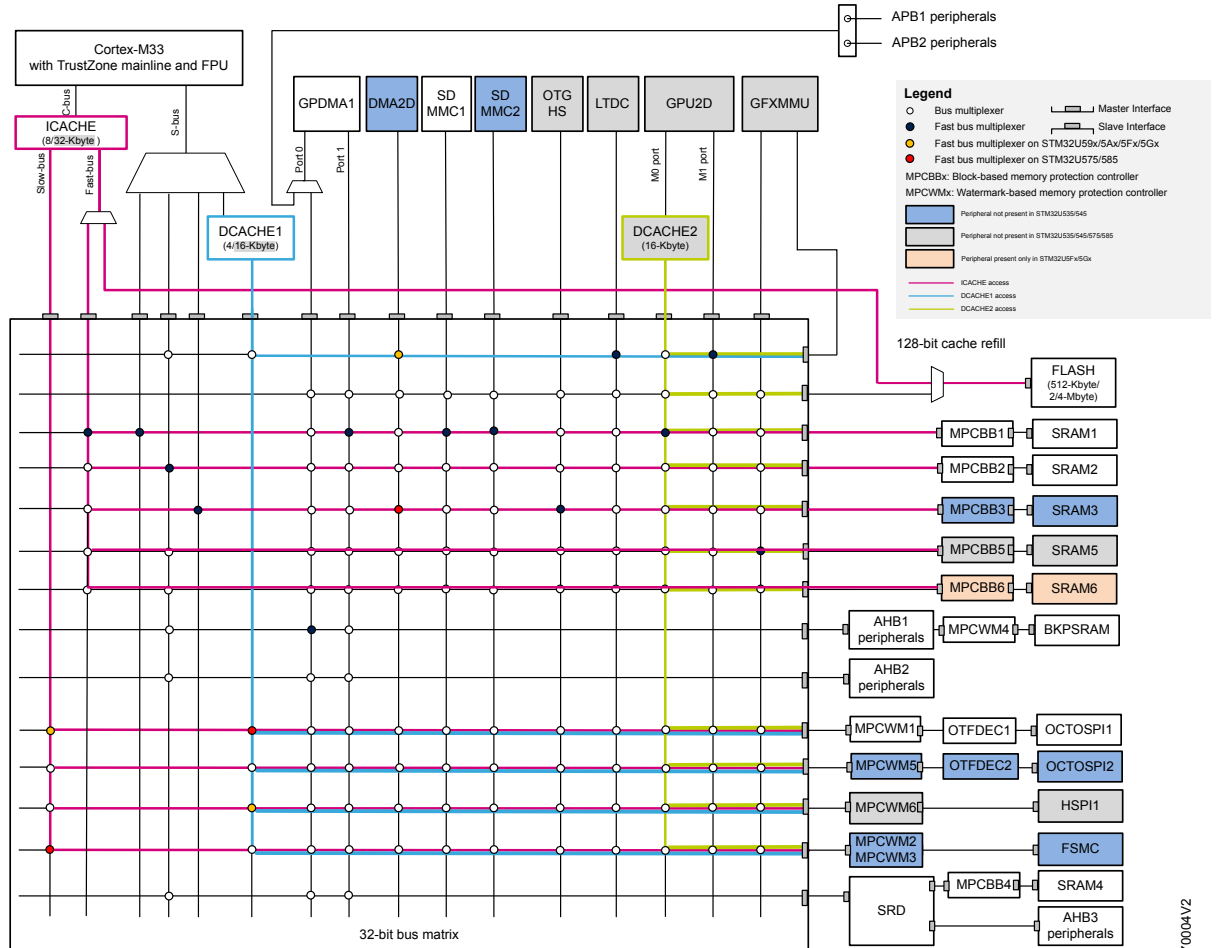
### 2.2 STM32U5 series smart architecture

The architecture is based on a bus matrix allowing multiple masters (Cortex®-M33, ICACHE, DCACHE, GPDMA, DMA2D, and SDMMCs, OTG\_HS, LTDC, GPU2D, GFXMMU) to access multiple slaves (such as flash memory, SRAMs, BKPSRAM, HSPI/OCTOSPI, or FSMC).

DT63699V1

The figure below describes the STM32U5 series smart architecture.

Figure 2. STM32U5 series smart architecture



The Cortex<sup>®</sup>-M33 and the GPU2D interfaces both benefit from using CACHE.

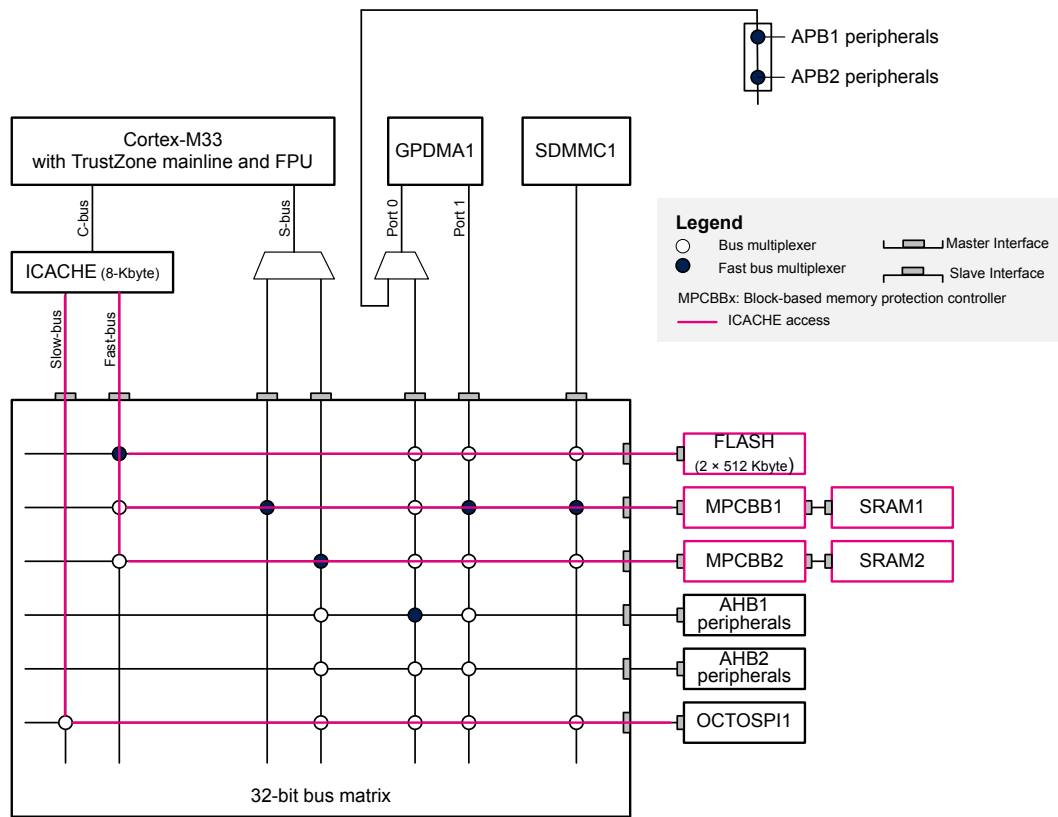
- ICACHE improves the performance of Cortex<sup>®</sup>-M33 when fetching code or data from the internal memories through fast bus (flash memory, SRAMs) and from external memories through slow bus (OCTOSPI1/2 and HSP1, or FSMC). DCACHE1 improves the performance when fetching data from internal or external memories through the s-bus (GFXMMU, OCTOSPI1/2 and HSP1, or FSMC).
- DCACHE2 improves the performance of GPU2D when fetching data from internal and external memories (GFXMMU, flash memory, SRAMs, OCTOSPI1/2, and HSP1, or FSMC) through the M0 port bus.

### 2.3 STM32U3 series smart architecture

The architecture is based on a bus matrix allowing multiple masters (Cortex<sup>®</sup>-M33, ICACHE, GPDMA1, and SDMMC1) to access multiple slaves (such as flash memory, SRAM1/2, or OCTOSPI1).

The figure below describes the STM32U3 series smart architecture.

Figure 3. Example of STM32U3 smart architecture



DT77015V1

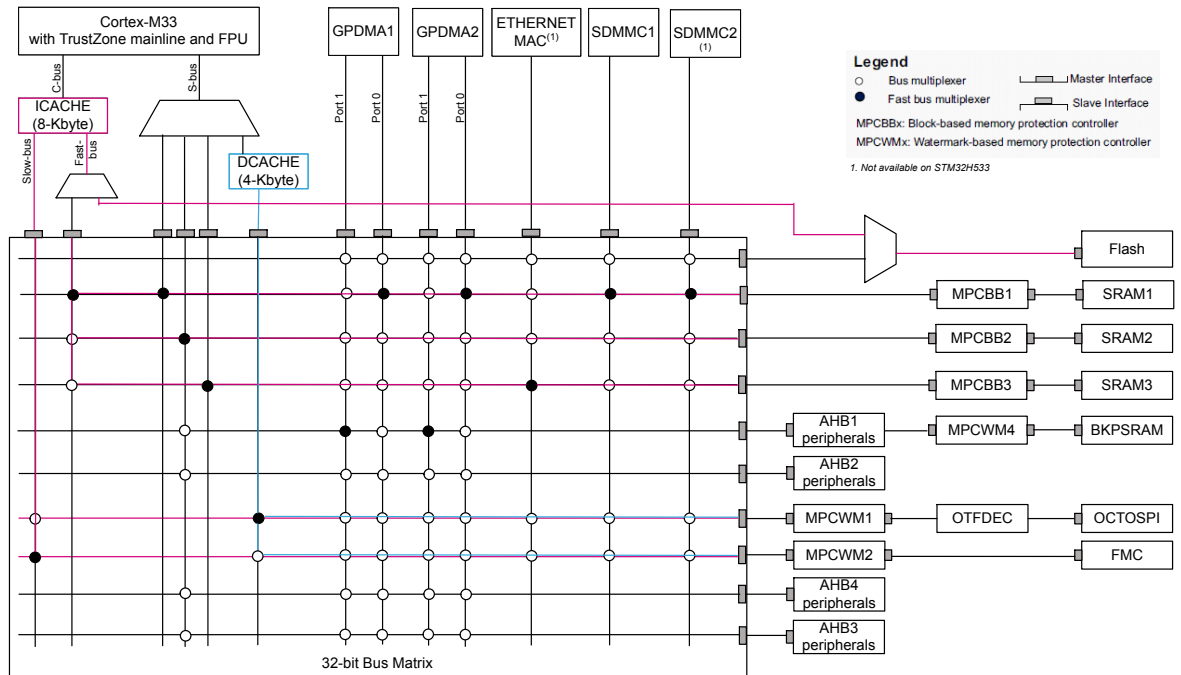
The ICACHE is introduced on the C-AHB code bus of the Cortex<sup>®</sup>-M33 processor to improve performance when fetching code or data from the internal memories through the fast bus (flash memory, SRAMs) and from external memories through the slow bus (OCTOSPI).

## 2.4 STM32H5 series smart architecture

The architecture is based on a bus matrix allowing multiple masters (Cortex<sup>®</sup>-M33, ICACHE, DCACHE, GPDMA, Ethernet, and SDMMCs) to access multiple slaves (such as flash memory, SRAMs, BKPSRAM, OCTOSPI, and FMC).

The figure below describes the STM32H5 series smart architecture.

Figure 4. Example of STM32H5 smart architecture



The Cortex<sup>®</sup>-M33 benefits from using CACHE.

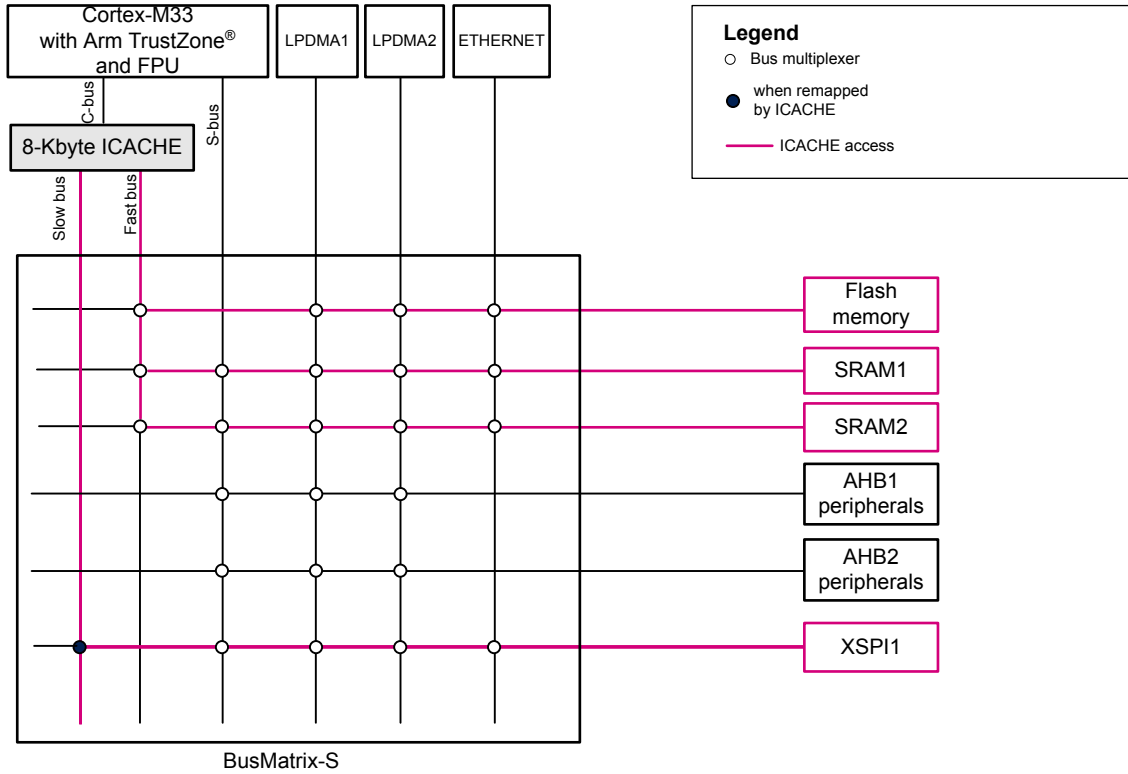
- ICACHE improves the performance of Cortex<sup>®</sup>-M33 when fetching code or data from the internal memories through the fast bus (flash memory, SRAMs) and from external memories through the slow bus (OCTOSPI and FMC).
- DCACHE improves the performance when fetching data from external memories through the slow bus (OCTOSPI and FMC).

## 2.5 STM32C5 series smart architecture

The architecture is based on a bus matrix allowing multiple masters (Cortex<sup>®</sup>-M33, ICACHE, LPDMA1/2, and ETHERNET) to access multiple slaves (such as flash memory, SRAM1/2 or XSPI1).

The figure below describes the STM32C5 series smart architecture.

Figure 5. STM32C5 series smart architecture



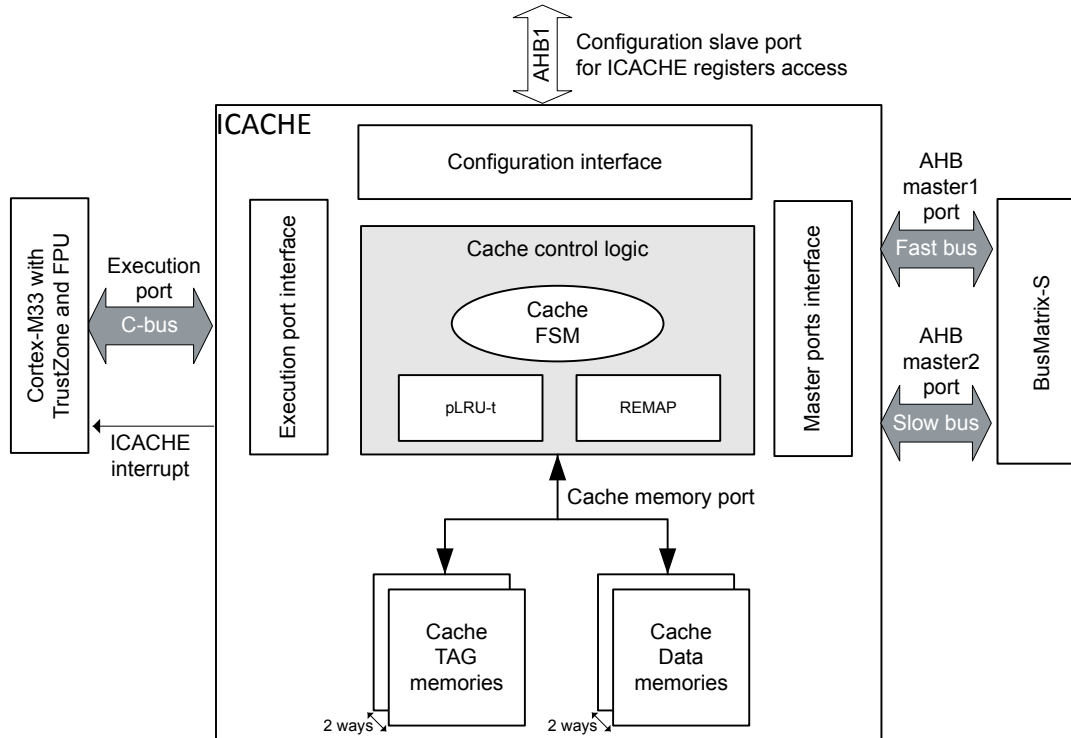
DT80407V1

The Cortex®-M33 performance is improved by using the 8-Kbyte ICACHE interface introduced to its C-AHB bus, when fetching code or data from the internal memories (flash memory, SRAM1, or SRAM2) through the fast bus, and also from the external memories (XSPI1) through the slow bus.

## 2.6 ICACHE block diagram

The ICACHE block diagram is given in the figure below.

Figure 6. ICACHE block diagram



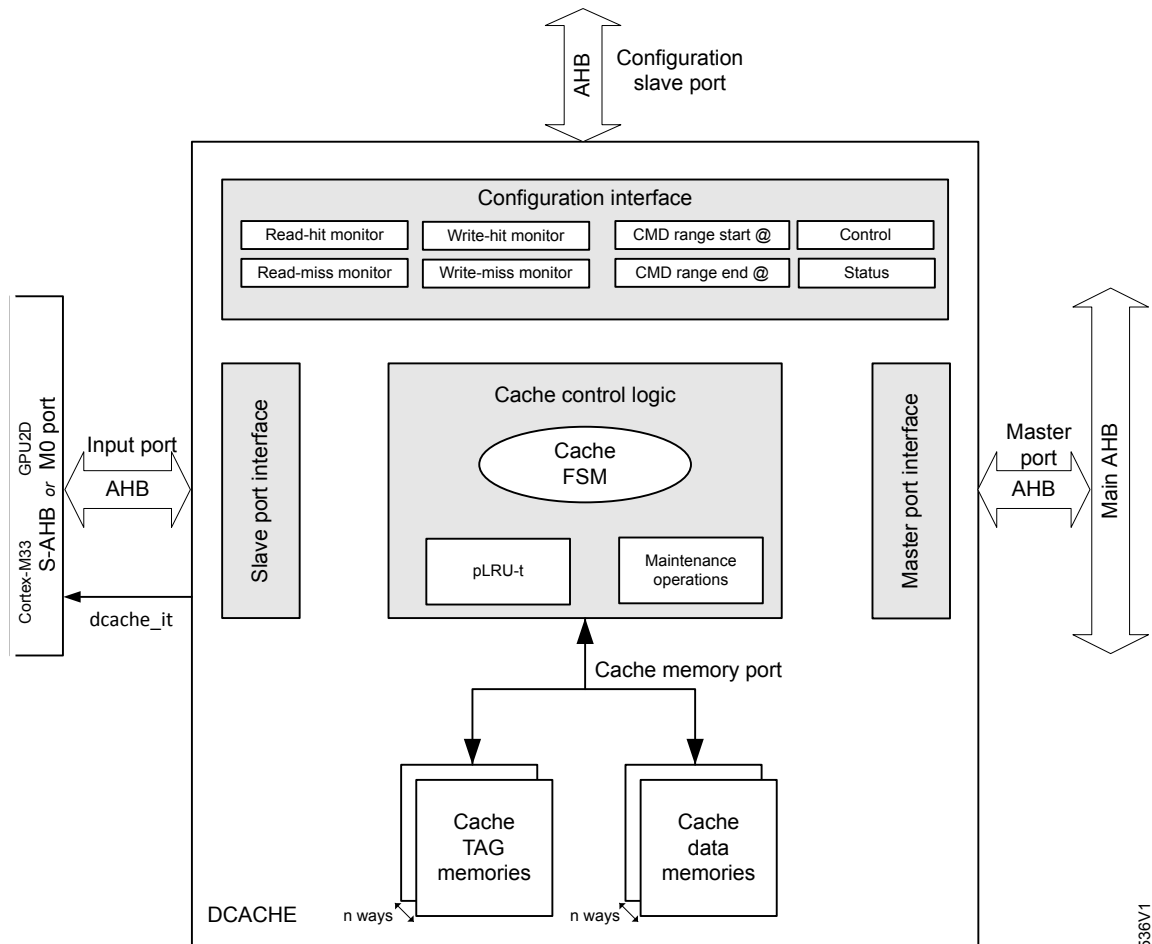
The ICACHE memory includes:

- the TAG memory with:
  - the address tags that indicate which data are contained in the cache data memory
  - the validity bits
- the data memory, that contains the cached data

## 2.7 DCACHE block diagram

The DCACHE block diagram is given in the figure below.

Figure 7. DCACHE block diagram



DT71536V1

The DCACHE memory includes:

- the TAG memory with:
  - the address tags that indicate which data are contained in the cache data memory
  - the validity bits
  - the privilege bits
  - the dirty bits
- the data memory, that contains the cached data

### 3 ICACHE and DCACHE features

#### 3.1 ICACHE features

##### 3.1.1 Dual masters

The ICACHE accesses the AHB bus matrix either over:

- One AHB master port: master1 (fast bus)
- Two AHB master ports: master1 (fast bus) and master2 (slow bus)

This feature allows the traffic to be decoupled when accessing different memory regions (such as internal flash memory, internal SRAM, and external memories), in order to reduce the CPU stalls on cache misses.

The following table summarizes memory regions and their addresses.

**Table 2. Memory regions and their addresses**

Peripheral				Cacheable memory access			Not cacheable memory access		
Type	Name	Product name and region size		Bus name	Nonsecure region starting address	Secure, nonsecure callable region starting address	Bus name	Nonsecure region starting address	Secure, nonsecure callable region starting address
Internal	FLASH	STM32H503	128 KB	ICACHE fast bus	0x0800 0000	N/A	N/A	N/A	N/A
		STM32C53x/ C54x	256 KB			N/A			
		STM32L5 series/ STM32U535/ 545/ STM32H523/ 533 STM32C55x/ C56x	512 KB			N/A			
		STM32U375/ 385 STM32C59x/ C5Ax	1 MB			0x0C00 0000 (1)			
		STM32U575/ 585 STM32H563/ 573/562	2 MB			N/A			
		STM32U59x/ 5Ax/5Fx/5Gx	4 MB			N/A			
		STM32H503	16 KB			N/A			
	SRAM1	STM32L5 series STM32U375/ 385 STM32U535/ 545/575/585	192 KB	0x0A00 0000	0x0E00 0000	S-bus	0x2000 0000	0x3000 0000	
		STM32H523/ 533	128 KB						
		STM32H563/ 573/562	256 KB						

Peripheral				Cacheable memory access			Not cacheable memory access		
Internal	SRAM1	STM32U59x/ 5Ax/5Fx/5Gx	768 KB	ICACHE fast bus	0x0A00 0000	0x0E00 0000	0x2000 0000	0x3000 0000	
		STM32C53x/ C54x	32 KB						
		STM32C55x/ C56x	64 KB						
		STM32C59x/ C5Ax	128 KB						
	SRAM2	STM32C53x/ C54x	32 KB		0x0A00 8000	N/A	0x2000 8000	N/A	
		STM32C55x/ C56x	64 KB		0x0A01 0000		0x2001 0000		
		STM32C59x/ C5Ax	128 KB		0x0A02 0000		0x2002 0000		
		STM32H503 series	16 KB		0x0A00 4000		0x2000 4000		
		STM32L5 series STM32U535/ 545/575/585	64 KB		0x0A03 0000	0x0E03 0000	0x2003 0000	0x3003 0000	
		STM32H523/ 533	64 KB		0x0A04 0000	0x0E04 0000	0x2004 0000	0x3004 0000	
		STM32H563/ 573/562	80 KB						
		STM32U375/ 385	64 KB		0x0A03 0000	0x0E03 0000	0x2003 0000	0x3003 0000	
		STM32U59x/ 5Ax/5Fx/5Gx	64 KB		0x0A0C 0000	0x0E0C 0000	0x200C 0000	0x300C 0000	
		SRAM3	STM32U575/ 585		512 KB	0x0A04 0000	0x0E04 0000	0x2004 0000	0x3004 0000
	STM32H523/ 533		64 KB		0x0A05 0000	0x0E05 0000	0x2005 0000	0x3005 0000	
	STM32H563/ 573/562		320 KB						
	STM32U59x/ 5Ax/5Fx/5Gx		832 KB						
	STM32U59x/ 5Ax/5Fx/5Gx	832 KB	0x0A1A 0000						0x0E1A 0000
SRAM5	STM32U5Fx/ 5Gx	512 KB	0x0A27 0000	0x0E27 0000	0x2027 0000				
External	HSPI1	STM32U59x/ 5Ax/5Fx/5Gx	256 MB	ICACHE slow bus (2)	Alias address in the range of [0x0000 0000 to 0x07FF FFFF] or [0x1000 0000:0x1FFF FFFF] defined by means of remapping feature	N/A	0xA000 0000	N/A	
	FMC SDRAM	STM32H563/ 573/562					0xC000 0000		
	OCTOSPI1 bank nonsecure	STM32U375/ 385 STM32L5/U5 series STM32H563/ 573/562 STM32C59x/ C5Ax					0x9000 0000		

S-bus

Peripheral			Cacheable memory access			Not cacheable memory access			
External	FMC bank 3 nonsecure	STM32L5/U5 series STM32H563/573/562	256 MB	ICACHE slow bus <sup>(2)</sup>	Alias address in the range of [0x0000 0000 to 0x07FF FFFF] or [0x1000 0000:0x1FFF FFFF] defined by means of remapping feature	N/A	S-bus	0x8000 0000	N/A
	OCTOSPI2 bank nonsecure	STM32U575/585/59x/5Ax/5Fx/5Gx						0x7000 0000	
	FMC bank 1 nonsecure	STM32L5/U5 series STM32H563/573/562						0x6000 0000	

1. N/A for STM32C5xx

2. To be selected when remapping such regions.

### 3.1.2 1-way versus 2-way ICACHE

By default, the ICACHE is configured in associative operating mode (two ways enabled), but it is possible to configure the ICACHE in direct mapped mode (one way enabled), for applications requiring a very-low power consumption. The ICACHE configuration is done with the WAYSEL bit in ICACHE\_CR as follows:

- WAYSEL = 0: direct mapped operating mode (1-way)
- WAYSEL = 1 (default): associative operating mode (2-way)

**Table 3. 1-way versus 2-way ICACHE**

Parameter	1-way ICACHE	2-way ICACHE
Cache size (Kbytes)	8 <sup>(1)</sup> /32 <sup>(2)</sup>	
Cache number of ways	1	2
Cache line size	128 bits (16 bytes)	
Number of cache lines	512 <sup>(1)</sup> /2048 <sup>(2)</sup>	256 <sup>(1)</sup> /1024 <sup>(2)</sup> per way

1. For STM32L5 series, STM32H5 series STM32U535/545/575/585, and STM32U3 series

2. For STM32U59x/5Ax/5Fx/5Gx

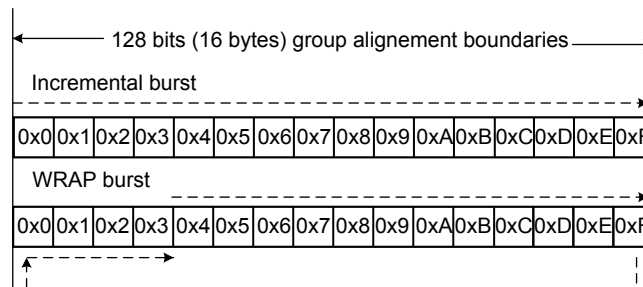
### 3.1.3 Burst type

Some Octo-SPI memories support the WRAP burst, that provides the benefit of critical-word-first feature performance. The ICACHE burst type of the AHB memory transaction for remapped regions is configurable. It implements incremental burst or WRAP burst, selected with the HBURST bit in the ICACHE\_CRRx register.

The differences between the WRAP and the incremental bursts are given below (see also the figure):

- WRAP burst:
  - cache line size = 128 bits
  - burst starting address = word address of the first data requested by the CPU
- Incremental burst:
  - cache line size = 128 bits
  - burst starting address = address aligned on the boundary of the cache line containing the requested word

Figure 8. Incremental versus WRAP burst



### 3.1.4 Cacheable regions and remapping feature

The ICACHE is connected to the Cortex<sup>®</sup>-M33 through the C-AHB bus, and caches the code region from addresses [0x0000 0000 to 0x1FFF FFFF].

Since the external memories are mapped at an address in the range [0x6000 0000 to 0xAFFF FFFF], the ICACHE supports a remap feature that allows any external memory region to be remapped at an address in the range of [0x0000 0000 to 0x07FF FFFF] or [0x1000 0000 to 0x1FFF FFFF], and to become accessible through the C-AHB bus.

Up to four external memory regions can be remapped with this feature.

Once a region is remapped, the remap operation occurs even if the ICACHE is disabled or if the transaction is not cacheable.

The cacheable memory regions can be defined and programmed by the user in the memory protection unit (MPU). The table below summarizes the configurations of the STM32L5, STM32U3, and STM32U5 series memories.

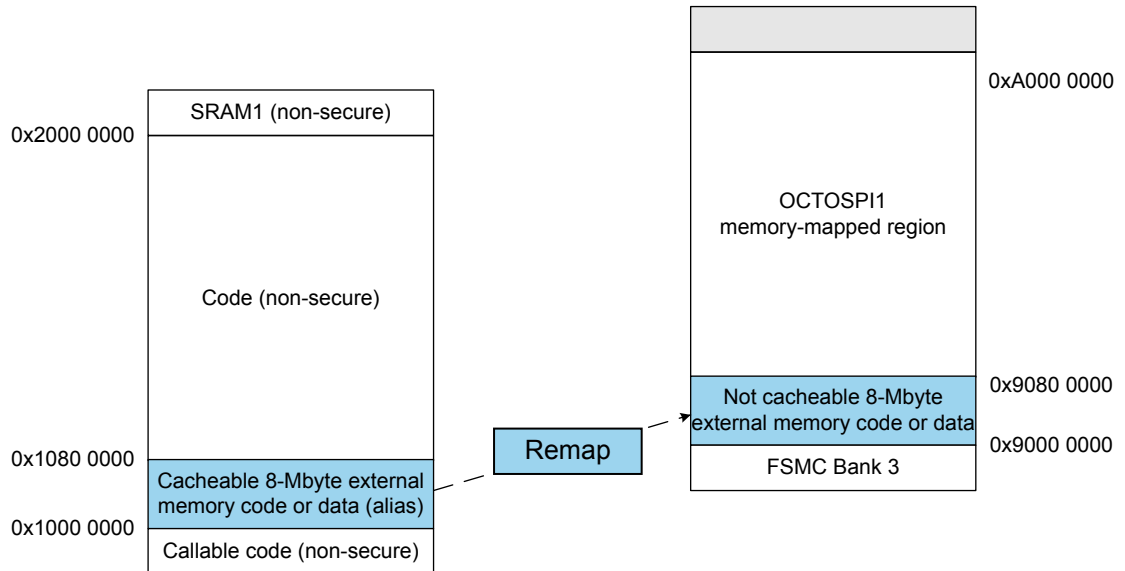
Table 4. Configuration of STM32L5, STM32U3, and STM32U5 series memories

Product memory	Cacheable (MPU programming)	Remapped in ICACHE (ICACHE_CRRx programming)
Flash memory	Yes or No	Not required
SRAM	Not recommended	
External memories (HSPI/OCTOSPI or FSMC)	Yes or No	Required if the user wants external code fetching on C-AHB bus (else on S-AHB bus)

### 3.1.5 Benefit of ICACHE external memory remapping

The example in the figure below shows how to benefit from the ICACHE enhanced performance during code execution or data read when accessing an external 8-Mbyte external Octo-SPI memory (such as external flash memory or RAM).

Figure 9. Octo-SPI memory remap example



The following steps are needed to remap this external memory:

1. OCTOSPI configuration for the external memory  
Configure the OCTOSPI interface in order to access the external memory in Memory mapped mode (the external memory is seen as an internal memory mapped in the [0x9000 0000 to 0x9FFF FFFF] region). Since the external memory size is 8 Mbytes, it is seen at the region [0x9000 0000 to 0x907F FFFF]. The external memory at this region is accessed through the S-bus and is not cacheable. The next step shows the ICACHE configuration in order to remap this region.

*Note:* For the OCTOSPI configuration in memory-mapped mode, refer to the application note Octo-SPI interface on STM32 microcontrollers (AN5050).

2. ICACHE configuration to remap the external memory mapped region
 

The 8 Mbytes placed in the [0x9000 0000 to 0x907F FFFF] region are remapped to the [0x1000 0000 to 0x107F FFFF] region. They can then be accessed through the slow bus (ICACHE master2 bus).

  - ICACHE\_CR register configuration
    - a. Disable ICACHE with EN = 0.
    - b. Select 1-way or 2-ways (depending on the application needs) with WAYSEL = 0 or 1, respectively.
  - ICACHE\_CRRx register configuration (up to four regions, x = 0 to 3)
    - a. Select the 0x1000 0000 base address (remap address) with BASEADDR [28:21] = 0x80.
    - b. Select the 8-Mbyte region size to remap with RSIZE[2:0] = 0x3.
    - c. Select the 0x9000 0000 remapped address REMAPADDR[31:21] = 0x480.
    - d. Select the ICACHE AHB master2 port for external memories with MSTSEL = 1.
    - e. Select the WRAP burst type with HBURST = 0.
    - f. Enable the remapping for region x with REN = 1.

The following figure shows how the memory regions are seen with IAR after enabling the remap.

**Figure 10. Memory regions remapping example**

Memory 1	Memory 2
Go to 0x90000000	Go to 0x10000000
0x90000000 20 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70 ****Memory-rapp	0x10000000 20 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70 ****Memory-rapp
0x90000010 85 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63 ed OSPI communic	0x10000010 85 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63 ed OSPI communic
0x90000020 61 74 69 6f 6e 2a 2a 2a 2a 20 20 2a 2a 2a 2a 2a 4d ation**** **	0x10000020 61 74 69 6f 6e 2a 2a 2a 2a 20 20 2a 2a 2a 2a 2a 4d ation**** **
0x90000030 65 6d 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 ory-rapped OSF	0x10000030 65 6d 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 ory-rapped OSF
0x90000040 49 20 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a I communication*	0x10000040 49 20 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a I communication*
0x90000050 2a 2a 2a 20 2a 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d *** **Memory-	0x10000050 2a 2a 2a 20 2a 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d *** **Memory-
0x90000060 6d 61 70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d rapped OSPI comm	0x10000060 6d 61 70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d rapped OSPI comm
0x90000070 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a 20 20 2a unication**** *	0x10000070 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a 20 20 2a unication**** *
0x90000080 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70 65 64 ***Memory-rapped	0x10000080 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70 65 64 ***Memory-rapped
0x90000090 20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63 61 74 OSPI communicat	0x10000090 20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63 61 74 OSPI communicat
0x900000a0 69 6f 6e 2a 2a 2a 20 20 2a 2a 2a 4d 65 6d ion**** **New	0x100000a0 69 6f 6e 2a 2a 2a 20 20 2a 2a 2a 4d 65 6d ion**** **New
0x900000b0 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 49 20 ory-rapped OSPI	0x100000b0 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 49 20 ory-rapped OSPI
0x900000c0 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a unication****	0x100000c0 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a unication****
0x900000d0 2a 20 20 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 * **Memory-ra	0x100000d0 2a 20 20 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 * **Memory-ra
0x900000e0 70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e pped OSPI commu	0x100000e0 70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e pped OSPI commu
0x900000f0 69 63 61 74 69 6f 6e 2a 2a 2a 20 05 d7 75 5d ication**** ..u]	0x100000f0 69 63 61 74 69 6f 6e 2a 2a 2a 20 05 d7 75 5d ication**** ..u]

The 8-Mbyte external memory is now remapped and can be accessed over the [0x1000 0000 to 0x107F FFFF] region.

3. ICACHE enable
  - ICACHE\_CR register configuration
 

Enable the ICACHE with EN = 1.

### 3.1.6 Hit and miss monitors

ICACHE provides two monitors for performance analysis: a 32-bit hit monitor and a 16-bit miss monitor.

- The hit monitor counts the cacheable AHB-transactions on slave cache port that hit ICACHE content (fetched data already available in the cache). The hit monitor counter is available in the ICACHE\_HMONR register.
- The miss monitor counts the cacheable AHB-transactions on slave cache port that miss ICACHE content (fetched data not already available in the cache). The miss monitor counter is available in the ICACHE\_MMONR register.

*Note:* These two monitors do not wrap over when reaching their maximum values.

These monitors are managed from the following bits in the ICACHE\_CR register:

- HITMEN bit (respectively MISSMEN bit) to enable/stop the hit (respectively miss) monitor
- HITMRST bit (respectively MISSMRST bit) to reset the hit (respectively miss) monitor

By default, these monitors are disabled in order to reduce power consumption.

### 3.1.7 ICACHE maintenance

The software can invalidate the ICACHE by setting the CACHEINV bit in the ICACHE\_CR register. This action invalidates the whole cache, making it empty. Meanwhile, if some remapped regions are enabled, the remap feature is still active, even when the ICACHE is disabled.

As the ICACHE only manages read transactions and does not manage write transactions, it does not ensure coherency in case of writes. Consequently, the software must invalidate the ICACHE after programming a region.

### 3.1.8 ICACHE security

ICACHE is a securable peripheral that can be configured as secure through the GTZC TZSC secure configuration register. When it is configured as secure, only secure accesses are allowed to the ICACHE registers. ICACHE can also be configured as privileged through the GTZC TZSC privilege configuration register. When ICACHE is configured as privileged, only privileged accesses are allowed to the ICACHE registers.

By default, the ICACHE is nonsecure and non-privileged through the GTZC TZSC.

### 3.1.9 Event and interrupt management

The ICACHE manages the functional errors when detected, by setting the ERRF flag in ICACHE\_SR. An interrupt can also be generated if the ERRIE bit is set in ICACHE\_IER.

In case of ICACHE invalidation, when the cache busy state finished, the BSYENDF flag is set in ICACHE\_SR. An interrupt can also be generated if the BSYENDIE bit is set in ICACHE\_IER.

The table below lists the ICACHE interrupt and event flags.

**Table 5. ICACHE interrupt and event management bits**

Register	Bit name	Bit description	Bit access type
ICACHE_SR	BUSYF	Cache executing a full invalidate operation	Read-only
	BSYENDF	Cache invalidation operation finished	
	ERRF	An error occurred during caching operation	
ICACHE_IER	ERRIE	Enable interrupt for cache error	Read/write
	BSYENDIE	Enable interrupt in case of invalidation operation finished	
ICACHE_FCR	CERRF	Clears ERRF in ICACHE_SR	Write-only
	CBSYENDF	Clears BSYENDF in ICACHE_SR	

## 3.2 DCACHE features

The purpose of the data cache is to cache external memories data loads and data stores coming from the processor or from another bus master peripheral. DCACHE manages both read and write transactions.

### 3.2.1 DCACHE cacheability traffic

The DCACHE caches the external memories from the master port interface through the AHB bus. The incoming memory requests are defined cacheable according to its AHB transaction memory lockup attribute.

The DCACHE write policy is defined as write-through or write-back depending on the memory attribute configured by the MPU. When a region is configured as noncacheable, the DCACHE is bypassed.

**Table 6. DCACHE cacheability for AHB transaction**

AHB lookup attribute	AHB bufferable attribute	Cacheability
0	X	Read and write: noncacheable
1	0	Read: cacheable Write: (cacheable) write-through
1	1	Read: cacheable Write: (cacheable) write-back

### 3.2.2 DCACHE cacheable regions

For the STM32U5 series, the DCACHE1 slave interface is connected to the Cortex<sup>®</sup>-M33 through the S-AHB bus, and caches the GFXMMU, FMC, and HSPI/OCTOSPIs. The DCACHE2 slave interface is connected to the DMA2D through the M0 port bus, and caches all the internal and external memories (except SRAM4 and BRKPSRAM).

For the STM32H5 series, the DCACHE slave interface is connected to the Cortex®-M33 through the S-AHB external memories through FMC and OCTOSPI.

**Table 7. DCACHE cacheable regions and interfaces**

Cacheable memory address region	DCACHE1 cacheable interfaces	DCACHE2 cacheable interfaces
GFXMMU	X	X
SRAM1	N/A	X
SRAM2		X
SRAM3		X
SRAM5		X
SRAM6		X
HSPI1	X	X
OCTOSPI1	X	X
FMC BANKs	X	X
OCTOSPI2	X	X

*Note:* Some interfaces are not supported in certain products. Refer to [Figure 1](#) or the specific product reference manual.

### 3.2.3 Burst type

Same as ICACHE, the DCACHE supports incremental and wrapped bursts (see [Section 3.1.3](#)). For DCACHE, the burst type is configured through the HBURST bit in DCACHE\_CR.

### 3.2.4 DCACHE configuration

During boot, DCACHE is disabled by default making the slave memory requests forwarded directly to the master port. To enable DCACHE, EN bit must be set in the DCACHE\_CR register.

### 3.2.5 Hit and miss monitors

The DCACHE implements four monitors for cache performance analysis:

- Two 32-bit (R/W) hit monitors: count the number of times the CPU read or write data in the cache memory without generating a transaction on DCACHE master ports (data already available in the cache). The (R/W) hit monitors counters are available respectively in the DCACHE\_RHMONR and DCACHE\_WHMONR registers.
- Two 16-bit (R/W) miss monitors: count the number of times the CPU read or write data in the cache memory and generates a transaction on DCACHE master ports, in order to load the data from the memory region (fetched data not already available in the cache). The (R/W) miss monitors counters are available respectively in the DCACHE\_RMMONR and DCACHE\_WMMONR registers.

*Note:* These four monitors do not wrap over when reaching their maximum values. These monitors are managed from the following bits in the DCACHE\_CR register:

- WHITMEN bit (respectively WMISSMEN bit) to enable/stop the write hit (respectively miss) monitor
- RHITMEN bit (respectively RMISSMEN bit) to enable/stop the read hit (respectively miss) monitor
- WHITMRST bit (respectively WMISSMRST bit) to reset the write hit (respectively miss) monitor
- RHITMRST bit (respectively RMISSMRST bit) to reset the read hit (respectively miss) monitor

By default, these monitors are disabled in order to reduce power consumption.

### 3.2.6 DCACHE maintenance

The DCACHE offers multiple maintenance operations that can be configured through CACHECMD[2:0] in DCACHE\_CR.

000: no operation (default)

001: clean range. Clean a certain range in the cache  
 010: invalidate range. Invalidate a certain range in the cache  
 010: clean and invalidate range. Clean and invalidate a certain range in the cache

The selected range is configured through:

- CMDSTARTADDR register: command starting address
- CMDENDADDR register: command ending address

*Note:* This register must be set before CACHECMD is written.

The cache command maintenance starts when the STARTCMD bit is set in the DCACHE\_CR register. The DCACHE also support a full CACHE invalidation by setting the CACHEINV bit in the DCACHE\_CR register.

### 3.2.7 DCACHE security

The DCACHE is a securable peripheral that can be configured as secure through the GTZC TZSC secure configuration register. When it is configured as secure, only secure accesses are allowed to the DCACHE registers.

DCACHE can also be configured as privileged through the GTZC TZSC privilege configuration register. When DCACHE is configured as privileged, only privileged accesses are allowed to the DCACHE registers.

By default, the DCACHE is nonsecure and nonprivileged through the GTZC TZSC.

### 3.2.8 Event and interrupt management

The DCACHE manages the functional errors when detected, by setting the ERRF flag in DCACHE\_SR. An interrupt can also be generated if the ERRIE bit is set in DCACHE\_IER. In case of DCACHE invalidation, when the cache busy state finished, the BSYENDF flag is set in DCACHE\_SR.

An interrupt can also be generated if the BSYENDIE bit is set in DCACHE\_IER. The DCACHE command status can be checked through CMDENF and BUSYCMTDF through the DCACHE\_SR

An interrupt can also be generated if the CMDENDIE bit is set in DCACHE\_IER. The table below lists the DCACHE interrupts and events flags.

**Table 8. DCACHE Interrupt and events management bits**

Register	Register	Bit description	Bit access type
DCACHE_SR	BUSYF	Cache executing a full invalidate operation	Read-only
	BSYENDF	Cache full invalidate operation ended	
	BUSYCMTDF	Cache executing a range command	
	CMDENDF	A range command end	
	ERRF	An error occurred during caching operation	
DCACHE_IER	ERRIE	Enable interrupt for cache error	Read/write
	CMDENDIE	Enable interrupt on range command end	
	BSYENDIE	Enable interrupt on full invalidate operation end	
DCACHE_FCR	CERRF	Clears ERRF in DCACHE_SR	Write-only
	CCMDENDF	Clears CMDENDF in DCACHE_SR	
	CBSYENDF	Clears BSYENDF in DCACHE_SR	

## 4 ICACHE and DCACHE performance and power consumption

Using ICACHE and DCACHE improve the application performance when accessing external memories. The following table shows the impact of ICACHE and DCACHE on Cortex® execution when accessing external memories.

**Table 9. ICACHE and DCACHE performance on CoreMark execution with external memories**

(1)

CoreMark code	CoreMark data	ICACHE configuration	DCACHE configuration	CoreMark score/MHz
Internal flash memory	Internal SRAM	Enabled (2-ways)	Disabled	3.89
Internal flash memory	External octo-SPI PSRAM (S-bus)	Enabled (2-ways)	Enabled	3.89
Internal flash memory	External octo-SPI PSRAM (S-bus)	Enabled (2-ways)	Disabled	0.48
External octo-SPI flash (C-bus)	Internal SRAM	Enabled (2-ways)	Disabled	3.86
External octo-SPI flash (C-bus)	Internal SRAM	Disabled	Disabled	0.24
Internal flash memory	Internal SRAM	Disabled	Disabled	2.69

1. *Test conditions:*

- *Applicable product: STM32U575/585*
- *System frequency: 160 MHz.*
- *External octo-SPI PSRAM memory: 80 MHz (DTR mode).*
- *External octo-SPI flash memory: 80 MHz (STR mode).*
- *Compiler: IAR V8.50.4.*
- *Internal flash PREFETCH: ON.*

Using ICACHE and DCACHE reduce the power consumption when accessing internal and external memories. The following table shows the impact of ICACHE on power consumption during CoreMark execution.

**Table 10. CoreMark execution ICACHE impact on power consumption**

(1)

ICACHE configuration	MCU power consumption (mA)
Enabled (2-ways)	7.60
Enabled (1-way)	7.13
Disabled	8.89

1. *Test conditions:*

- *Applicable product: STM32U575/585*
- *CoreMark code: internal flash memory.*
- *CoreMark data: internal SRAM.*
- *Internal flash memory PREFETCH: ON.*
- *System frequency: 160 MHz.*
- *Compiler: IAR V8.32.2.*
- *Voltage range: 1.*
- *SMPS: ON.*

**Note:** *2-way set associative configuration is more performing than 1-way set associative configuration for code that cannot be fully loaded in cache. Meanwhile, 1-way set associative cache is almost always more power efficient than 2-way set associative cache. Each code has to be evaluated in both associativity configurations, in order to select the best trade-off between performance and power consumption. The selection depends on the user priority.*

## 5 Conclusion

---

The first caches developed by STMicroelectronics, ICACHE and DCACHE, are able to cache internal and external memories, offering performance enhancement for data traffic and instruction fetches. This document shows the different features supported by the ICACHE and DCACHE, their configuration simplicity and flexibility allow lower development cost and faster time to market.

## Revision history

Table 11. Document revision history

Date	Version	Changes
10-Oct-2019	1	Initial release.
27-Feb-2020	2	Updated: <ul style="list-style-type: none"> <li>Table 2. Memory regions and their addresses</li> <li>Section 2.1.7 ICACHE maintenance</li> <li>Section 2.1.8 ICACHE security</li> </ul>
7-Dec-2021	3	Updated: <ul style="list-style-type: none"> <li>Document title</li> <li>Introduction</li> <li>Section 1 ICACHE and DCACHE overview</li> <li>Section 4 Conclusion</li> </ul> Added: <ul style="list-style-type: none"> <li>Section 2 ICACHE and DCACHE features</li> <li>Section 3 ICACHE and DCACHE performance and power consumption</li> </ul>
15-Feb-2023	4	Updated: <ul style="list-style-type: none"> <li>Section 2.2: STM32U5 series smart architecture</li> <li>Section 2.7: DCACHE block diagram</li> <li>Section 3.1.1: Dual masters</li> <li>Section 3.1.2: 1-way versus 2-way ICACHE</li> <li>Section 3.1.4: Cacheable regions and remapping feature</li> <li>Section 3.2: DCACHE features</li> <li>Section 3.2.2: DCACHE cacheable regions</li> <li>Section 4: ICACHE and DCACHE performance and power consumption</li> </ul> Added: <ul style="list-style-type: none"> <li>Section 1: General information</li> </ul>
11-Mar-2024	5	Updated: <ul style="list-style-type: none"> <li>Section 2.4: STM32H5 series smart architecture</li> <li>Section 3.1.1: Dual masters</li> </ul>
17-May-2024	6	Updated: <ul style="list-style-type: none"> <li>Section 2.4: STM32H5 series smart architecture</li> </ul>
06-Feb-2025	7	Added: <ul style="list-style-type: none"> <li>Section 2.3: STM32U3 series smart architecture</li> </ul> Updated to add STM32U3 series: <ul style="list-style-type: none"> <li>Section Introduction</li> <li>Section 3.1.2: 1-way versus 2-way ICACHE</li> <li>Section 3.1.4: Cacheable regions and remapping feature</li> </ul>
13-01-2026	8	Added: <ul style="list-style-type: none"> <li>Section 2.5: STM32C5 series smart architecture</li> </ul> Updated to add STM32C5 series: <ul style="list-style-type: none"> <li>Table 1. Applicable products</li> <li>Table 2. Memory regions and their addresses</li> </ul> Updated document title.

## Contents

<b>1</b>	<b>General information</b>	<b>2</b>
<b>2</b>	<b>ICACHE and DCACHE overview</b>	<b>3</b>
2.1	STM32L5 series smart architecture	3
2.2	STM32U5 series smart architecture	3
2.3	STM32U3 series smart architecture	4
2.4	STM32H5 series smart architecture	5
2.5	STM32C5 series smart architecture	6
2.6	ICACHE block diagram	8
2.7	DCACHE block diagram	9
<b>3</b>	<b>ICACHE and DCACHE features</b>	<b>10</b>
3.1	ICACHE features	10
3.1.1	Dual masters	10
3.1.2	1-way versus 2-way ICACHE	12
3.1.3	Burst type	13
3.1.4	Cacheable regions and remapping feature	13
3.1.5	Benefit of ICACHE external memory remapping	14
3.1.6	Hit and miss monitors	15
3.1.7	ICACHE maintenance	15
3.1.8	ICACHE security	16
3.1.9	Event and interrupt management	16
3.2	DCACHE features	16
3.2.1	DCACHE cacheability traffic	16
3.2.2	DCACHE cacheable regions	16
3.2.3	Burst type	17
3.2.4	DCACHE configuration	17
3.2.5	Hit and miss monitors	17
3.2.6	DCACHE maintenance	17
3.2.7	DCACHE security	18
3.2.8	Event and interrupt management	18
<b>4</b>	<b>ICACHE and DCACHE performance and power consumption</b>	<b>19</b>
<b>5</b>	<b>Conclusion</b>	<b>20</b>
	<b>Revision history</b>	<b>21</b>
	<b>List of tables</b>	<b>23</b>
	<b>List of figures</b>	<b>24</b>

## List of tables

<b>Table 1.</b>	Applicable products . . . . .	1
<b>Table 2.</b>	Memory regions and their addresses . . . . .	10
<b>Table 3.</b>	1-way versus 2-way ICACHE. . . . .	12
<b>Table 4.</b>	Configuration of STM32L5, STM32U3, and STM32U5 series memories . . . . .	13
<b>Table 5.</b>	ICACHE interrupt and event management bits. . . . .	16
<b>Table 6.</b>	DCACHE cacheability for AHB transaction . . . . .	16
<b>Table 7.</b>	DCACHE cacheable regions and interfaces. . . . .	17
<b>Table 8.</b>	DCACHE Interrupt and events management bits . . . . .	18
<b>Table 9.</b>	ICACHE and DCACHE performance on CoreMark execution with external memories . . . . .	19
<b>Table 10.</b>	CoreMark execution ICACHE impact on power consumption . . . . .	19
<b>Table 11.</b>	Document revision history . . . . .	21

## List of figures

<b>Figure 1.</b>	STM32L5 series smart architecture . . . . .	3
<b>Figure 2.</b>	STM32U5 series smart architecture . . . . .	4
<b>Figure 3.</b>	Example of STM32U3 smart architecture . . . . .	5
<b>Figure 4.</b>	Example of STM32H5 smart architecture . . . . .	6
<b>Figure 5.</b>	STM32C5 series smart architecture . . . . .	7
<b>Figure 6.</b>	ICACHE block diagram . . . . .	8
<b>Figure 7.</b>	DCACHE block diagram . . . . .	9
<b>Figure 8.</b>	Incremental versus WRAP burst . . . . .	13
<b>Figure 9.</b>	Octo-SPI memory remap example . . . . .	14
<b>Figure 10.</b>	Memory regions remapping example . . . . .	15

**IMPORTANT NOTICE – READ CAREFULLY**

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers’ market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to [www.st.com/trademarks](http://www.st.com/trademarks). All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2026 STMicroelectronics – All rights reserved