Using STM32 cache to optimize performance and power efficiency

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®-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 and lines
Microcontrollers	STM32L5 Series, STM32U575/585 line

1 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.

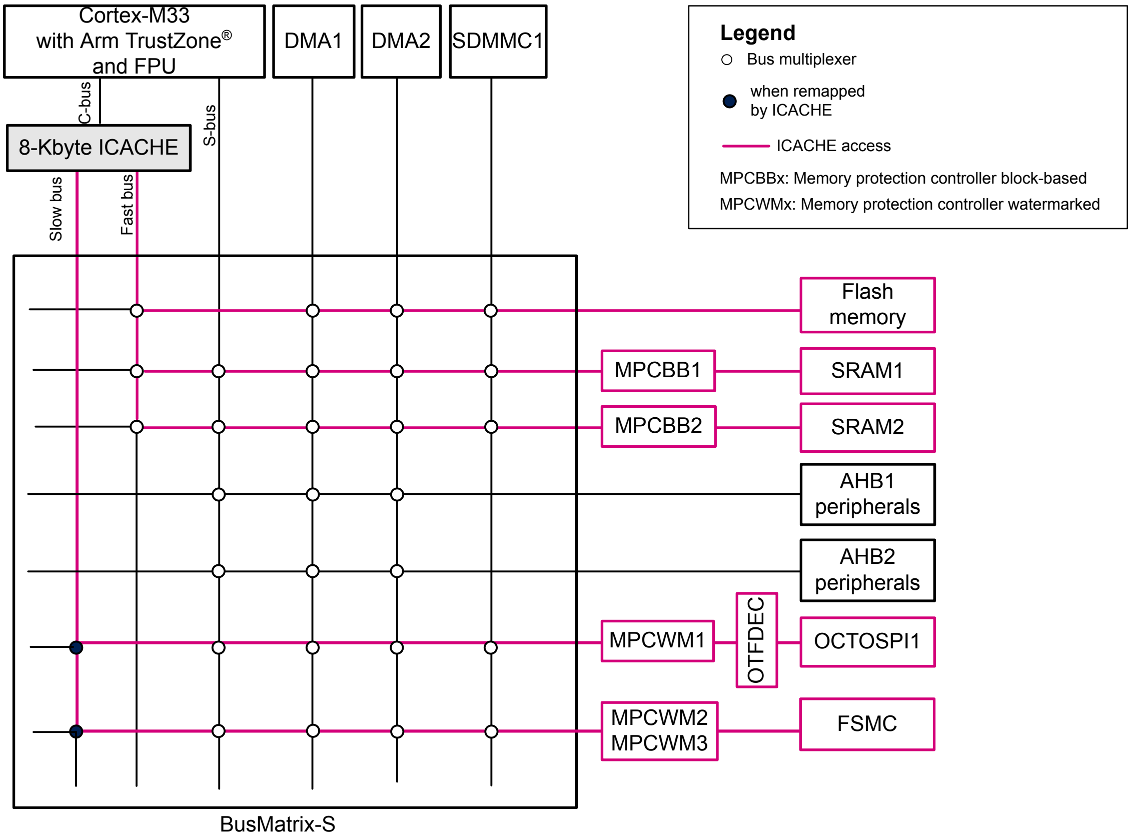
Note: Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

1.1 STM32L5 Series smart architecture

This 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



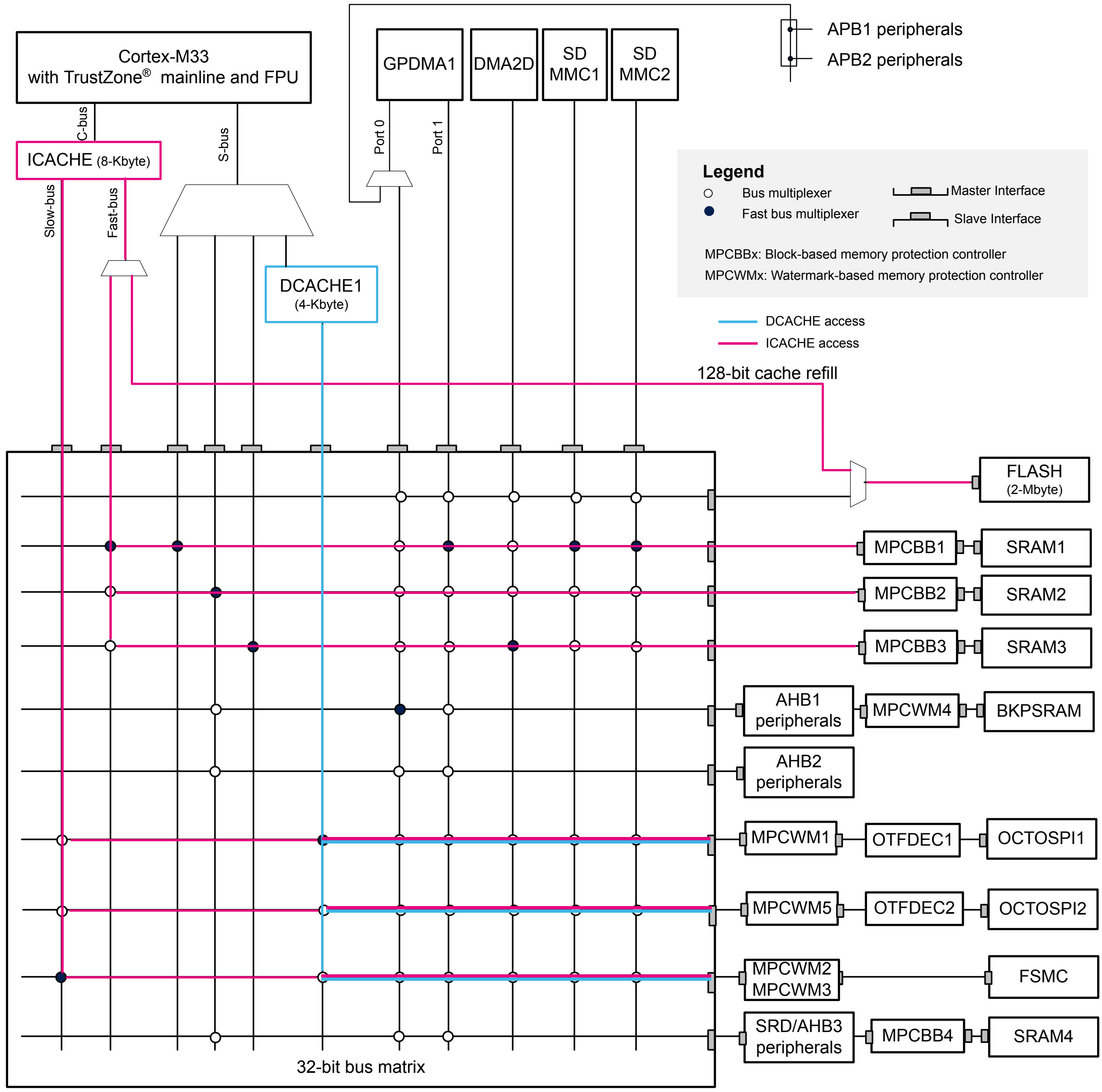
Thanks to the 8-Kbyte ICACHE interface introduced to its C-AHB bus, the Cortex-M33 performance is improved when fetching code or data from the internal memories (Flash memory, SRAM1 or SRAM2) via the Fast bus, and also from the external memories (OCTOSPI1 or FSMC) via the Slow bus.

1.2 STM32U5 Series smart architecture

This architecture is based on a bus matrix allowing multiple masters (Cortex-M33, ICACHE,DCACHE, GPDMA1 ,DMA2D and SDMMC1/2) to access multiple slaves (such as Flash memory, SRAM1/2/3/4, BKPSRAM, OCTOSPI1/2 or FSMC).

The figure below describes the STM32U5 Series smart architecture.

Figure 2. STM32U5 Series smart architecture



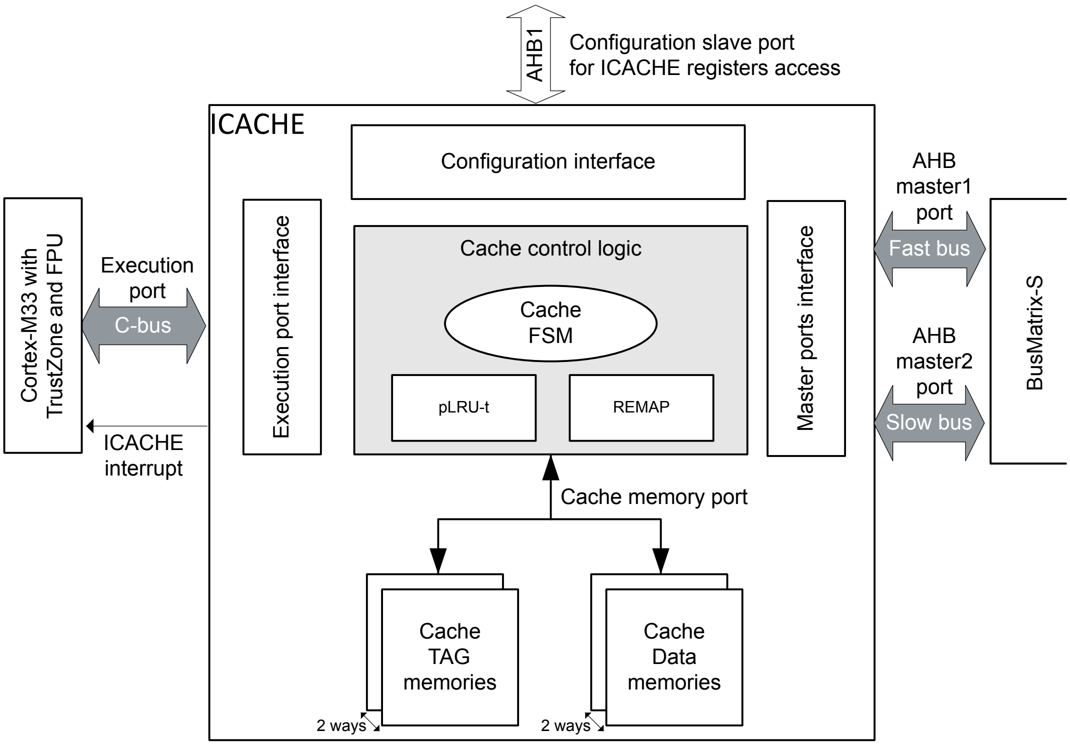
The 8-Kbyte ICACHE interface introduced to its C-AHB bus, improves the Cortex-M33 performance when fetching code or data from the internal memories (Flash memory, SRAM1,SRAM2 or SRAM3) via the Fast bus, and from the external memories (OCTOSPI1, OCTOSPI2 or FSMC) via the Slow bus.

The 4-Kbyte DCACHE interface introduced to its S-AHB bus, improves the data traffic from the external memories (OCTOSPI1, OCTOSPI2 or FSMC).

1.3 ICACHE block diagram

The ICACHE block diagram is given in the figure below.

Figure 3. ICACHE block diagram



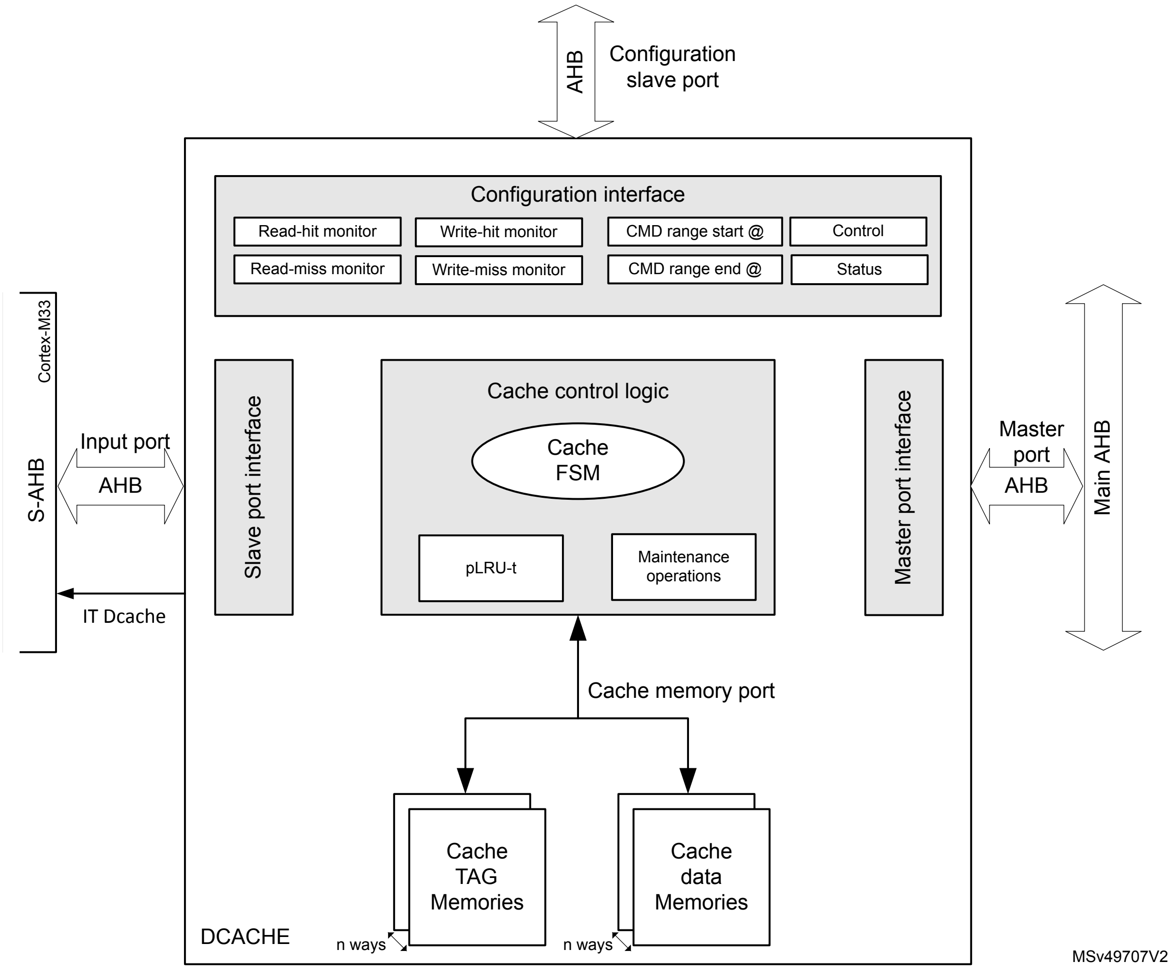
The ICACHE memory includes:

- 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

1.4 DCACHE block diagram

The DCACHE block diagram is given in the figure below.

Figure 4. DCACHE block diagram



The DCACHE memory includes:

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

2 ICACHE and DCACHE features

2.1 ICACHE features

2.1.1 Dual masters

The ICACHE accesses the AHB bus matrix over 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	Bus name	Region address	Bus name	Region address
Internal	FLASH	ICACHE Fast bus	[0x0800 0000 to 0x0807 FFFF] for L5 series [0x0800 0000 to 0x081F FFFF] for U5 series	N/A	N/A
	SRAM1		[0x0A00 0000 to 0x0A02 FFFF]	S-bus	[0x2000 0000 to 0x2002 FFFF]
	SRAM2		[0x0A03 0000 to 0x0A03 FFFF]		[0x2003 0000 to 0x2003 FFFF]
	SRAM3[1]		[0x0A04 0000 to 0x0A0B FFFF][1]		[0x2004 0000 to 0x200B FFFF][1]
External	OCTOSPI1 bank non-secure	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		[0x9000 0000 to 0x9FFF FFFF]
	FMC bank 3 non-secure				[0x8000 0000 to 0x8FFF FFFF]
	OCTOSPI2 bank non-secure [1]				[0x7000 0000 to 0x7FFF FFFF][1]
	FMC bank 1 non-secure				[0x6000 0000 to 0x6FFF FFFF]

1. *Not available for STM32L5 products.*
2. *To be selected when remapping such regions.*

2.1.2 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	
Cache number of ways	1	2
Cache line size	128 bits (16 bytes)	
Number of cache lines	512	256 per way

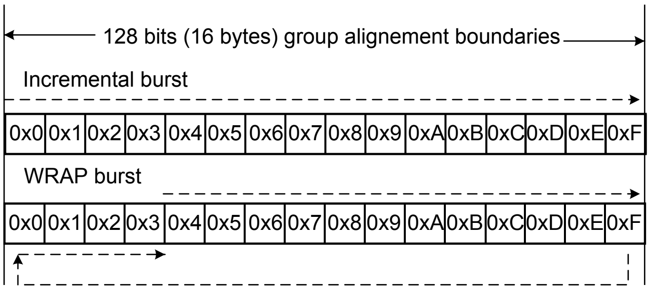
2.1.3 Burst type

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 5. Incremental versus WRAP burst


128 bits (16 bytes) group alignement boundaries
Incremental burst
0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF
WRAP burst
0x0 0x1 0x2 0x3 0x4 0x5 0x6 0x7 0x8 0x9 0xA 0xB 0xC 0xD 0xE 0xF

2.1.4 Cacheable regions and remapping feature

The ICACHE is connected to the Cortex-M33 via 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 0x9FFF 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 and STM32U5 Series memories.

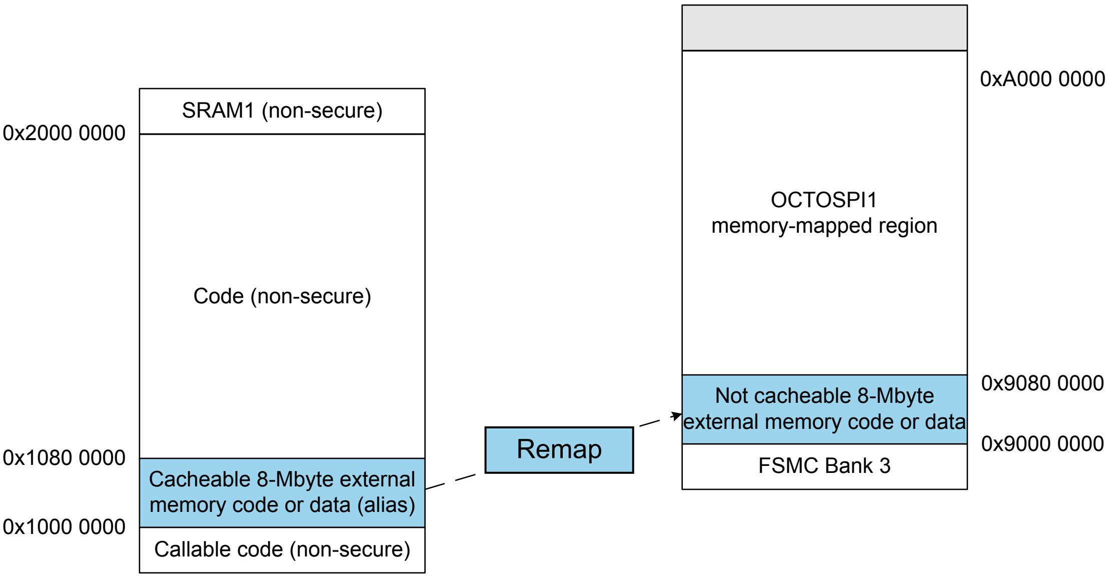
Table 4. Configuration of STM32L5 and STM32U5 Series memories

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

2.1.5 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 6. Octo-SPI memory remap example


SRAM1 (non-secure)
0x2000 0000
Code (non-secure)
0x1080 0000
Cacheable 8-Mbyte external memory code or data (alias)
0x1000 0000
Callable code (non-secure)
Remap
0xA000 0000
OCTOSPI1 memory-mapped region
0x9080 0000
Not cacheable 8-Mbyte external memory code or data
0x9000 0000
FSMC Bank 3

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 via 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.

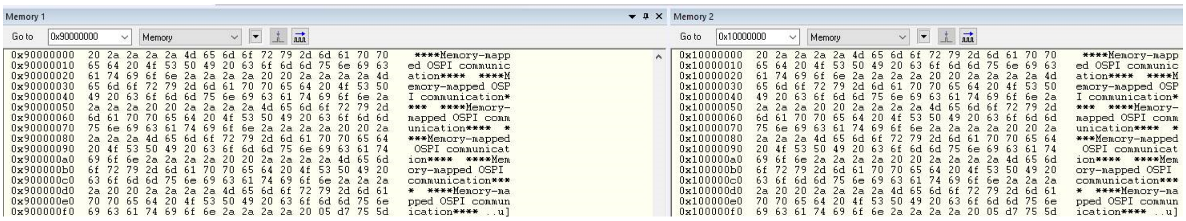
     d. Select the ICACHE AHB master2 port for external memories with MSTSEL = 1.

     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 7. Memory regions remapping example



```
Memory 1
Go to 0x90000000   Memory
0x90000000  20 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70   ****Memory-mapp
0x90000010  65 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 4d  ation****  ****M
0x90000030  65 6d 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50  emory-mapped OSP
0x90000040  49 20 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a  I communication*
0x90000050  2a 2a 2a 20 20 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  mapped OSPI comm
0x90000070  75 6e 69 63 61 74 69 6f 6e 2a 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-mapped
0x90000090  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 2a 20 20 2a 2a 2a 2a 4d 65 6d  ion****  ****Mem
0x900000b0  6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 49 20  ory-mapped OSPI
0x900000c0  63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a  communication***
0x900000d0  2a 20 20 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61  *  ****Memory-ma
0x900000e0  70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e  pped OSPI commun
0x900000f0  69 63 61 74 69 6f 6e 2a 2a 2a 2a 20 05 d7 75 5d  ication**** ..u]
```

```
Memory 2
Go to 0x10000000   Memory
0x10000000  20 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70   ****Memory-mapp
0x10000010  65 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63  ed OSPI communic
0x10000020  61 74 69 6f 6e 2a 2a 2a 2a 20 20 2a 2a 2a 2a 4d  ation****  ****M
0x10000030  65 6d 6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50  emory-mapped OSP
0x10000040  49 20 63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a  I communication*
0x10000050  2a 2a 2a 20 20 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d  ***  ****Memory-
0x10000060  6d 61 70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d  mapped OSPI comm
0x10000070  75 6e 69 63 61 74 69 6f 6e 2a 2a 2a 2a 20 20 2a  unication****  *
0x10000080  2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61 70 70 65 64  ***Memory-mapped
0x10000090  20 4f 53 50 49 20 63 6f 6d 6d 75 6e 69 63 61 74   OSPI communicat
0x100000a0  69 6f 6e 2a 2a 2a 2a 20 20 2a 2a 2a 2a 4d 65 6d  ion****  ****Mem
0x100000b0  6f 72 79 2d 6d 61 70 70 65 64 20 4f 53 50 49 20  ory-mapped OSPI
0x100000c0  63 6f 6d 6d 75 6e 69 63 61 74 69 6f 6e 2a 2a 2a  communication***
0x100000d0  2a 20 20 2a 2a 2a 2a 4d 65 6d 6f 72 79 2d 6d 61  *  ****Memory-ma
0x100000e0  70 70 65 64 20 4f 53 50 49 20 63 6f 6d 6d 75 6e  pped OSPI commun
0x100000f0  69 63 61 74 69 6f 6e 2a 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.

2.1.6 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, theses monitors are disabled in order to reduce power consumption.

2.1.7 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.

2.1.8 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 non-secure and non-privileged through the GTZC TZSC.

2.1.9 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	

2.2 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. These accesses include the instruction fetches that may occur at an external memory address. DCACHE manages both read and write transactions.

2.2.1 DCACHE cacheability traffic

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

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

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

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

2.2.2 DCACHE cacheable regions

The DCACHE slave interface is connected to the Cortex-M33 via the S-AHB bus, and caches the memory region from addresses [0x6000 0000:0x9FFF FFFF] for FMC, OCTOSPI1/OCTOSPI2.

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

Cacheable memory address region	Interface
[0x6000 0000:0x6FFF FFFF]	FMC bank 1 Non-secure
[0x7000 0000:0x7FFF FFFF]	OCTOSPI2 bank Non-secure
[0x8000 0000:0x8FFF FFFF]	FMC bank 3 Non-secure
[0x9000 0000:0x9FFF FFFF]	OCTOSPI1 bank Non-secure

2.2.3 Burst type

Same as ICACHE, the DCACHE supports incremental and wrapped bursts (see Section 2.1.3).

For DCACHE, the burst type is configured through the HBURST bit in DCACHE_CR.

2.2.4 DCACHE configuration

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

2.2.5 Hit and miss monitors

The DCACHE implements four monitors for cache performance analysis:

- Two 32-bit (R/W) hit monitor: counts the number of time 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 monitor: counts the number of time 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, theses monitors are disabled in order to reduce power consumption.

2.2.6 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 STARTCMD bit is set in DCACHE_CR register. The DCACHE also support a full CACHE invalidation by setting the CACHEINV bit in DCACHE_CR register.

2.2.7 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 non-secure and non- privileged through the GTZC TZSC.

2.2.8 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 BUSYCMDF 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	
	BUSYCMDF	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	

3 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 CoreMark® 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:*
   - *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:*
   - *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.

4 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: • Table 2. Memory regions and their addresses • Section 2.1.7 ICACHE maintenance • Section 2.1.8 ICACHE security
7-Dec-2021	3	Updated: • Document title • Introduction • Section 1 ICACHE and DCACHE overview • Section 4 Conclusion • Added • Section 2 ICACHE and DCACHE features • Section 3 ICACHE and DCACHE performance and power consumption

Contents

List of tables

List of figures

IMPORTANT NOTICE – PLEASE 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. 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 acknowledgement.

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 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.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, please refer to 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.

© 2021 STMicroelectronics – All rights reserved