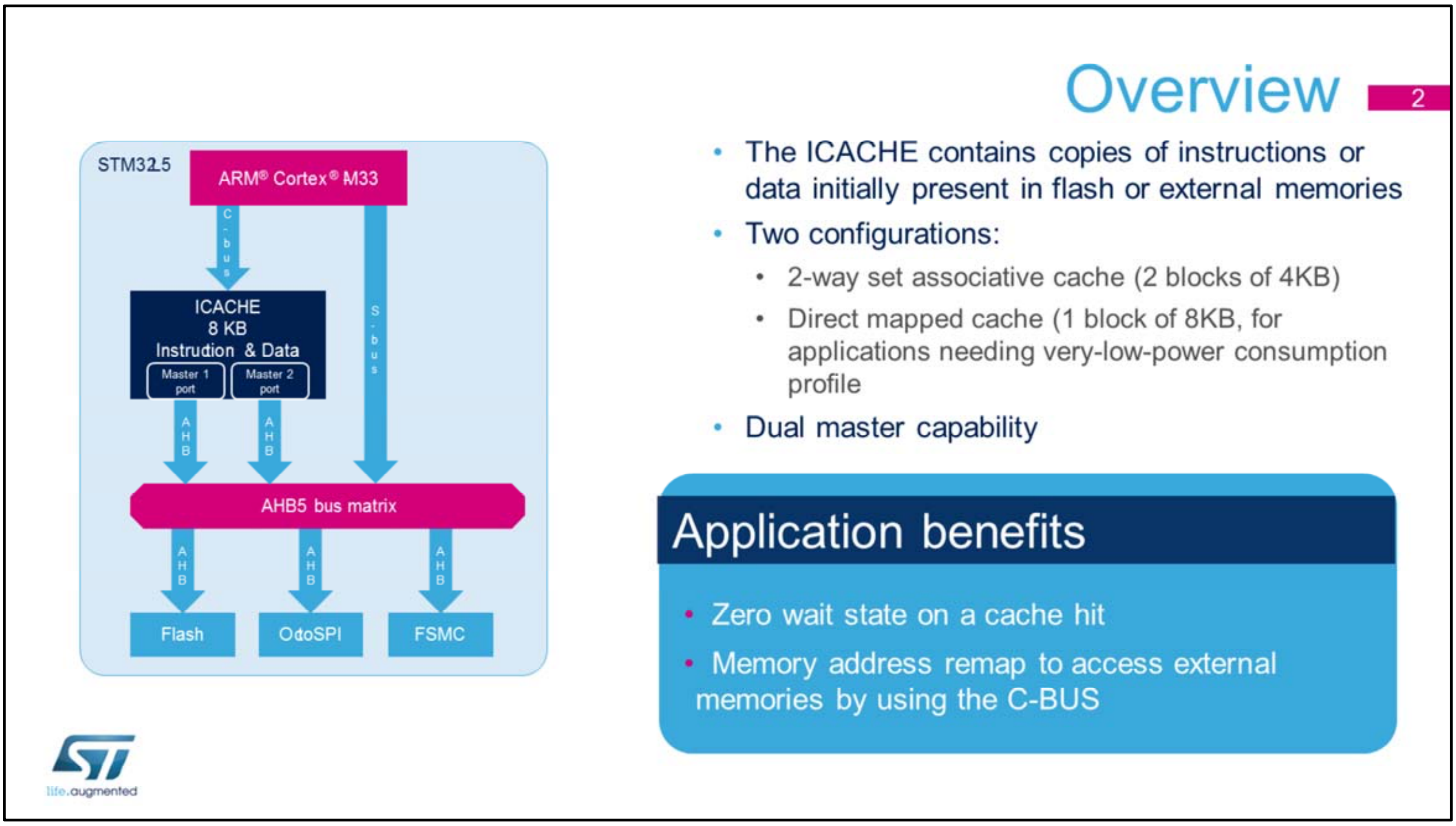
STM32L5- ICACHE

Instruction cache

Revision 1.0

life.augmented

Hello, and welcome to this presentation of the ICACHE module which is embedded in all products of the STM32L5 microcontroller family.



Overview

2

- The ICACHE contains copies of instructions or data initially present in flash or external memories
- Two configurations:
  - 2-way set associative cache (2 blocks of 4KB)
  - Direct mapped cache (1 block of 8KB, for applications needing very-low-power consumption profile
- Dual master capability

Application benefits

- Zero wait state on a cache hit
- Memory address remap to access external memories by using the C-BUS

The instruction cache (ICACHE) is introduced on the C-AHB code bus of Cortex®-M33 processor to improve performance when fetching instructions and data from both internal and external memories.

It implements a slave port connected to the Cortex-M33 C-BUS and two master ports connected to the AHB5 bus matrix.

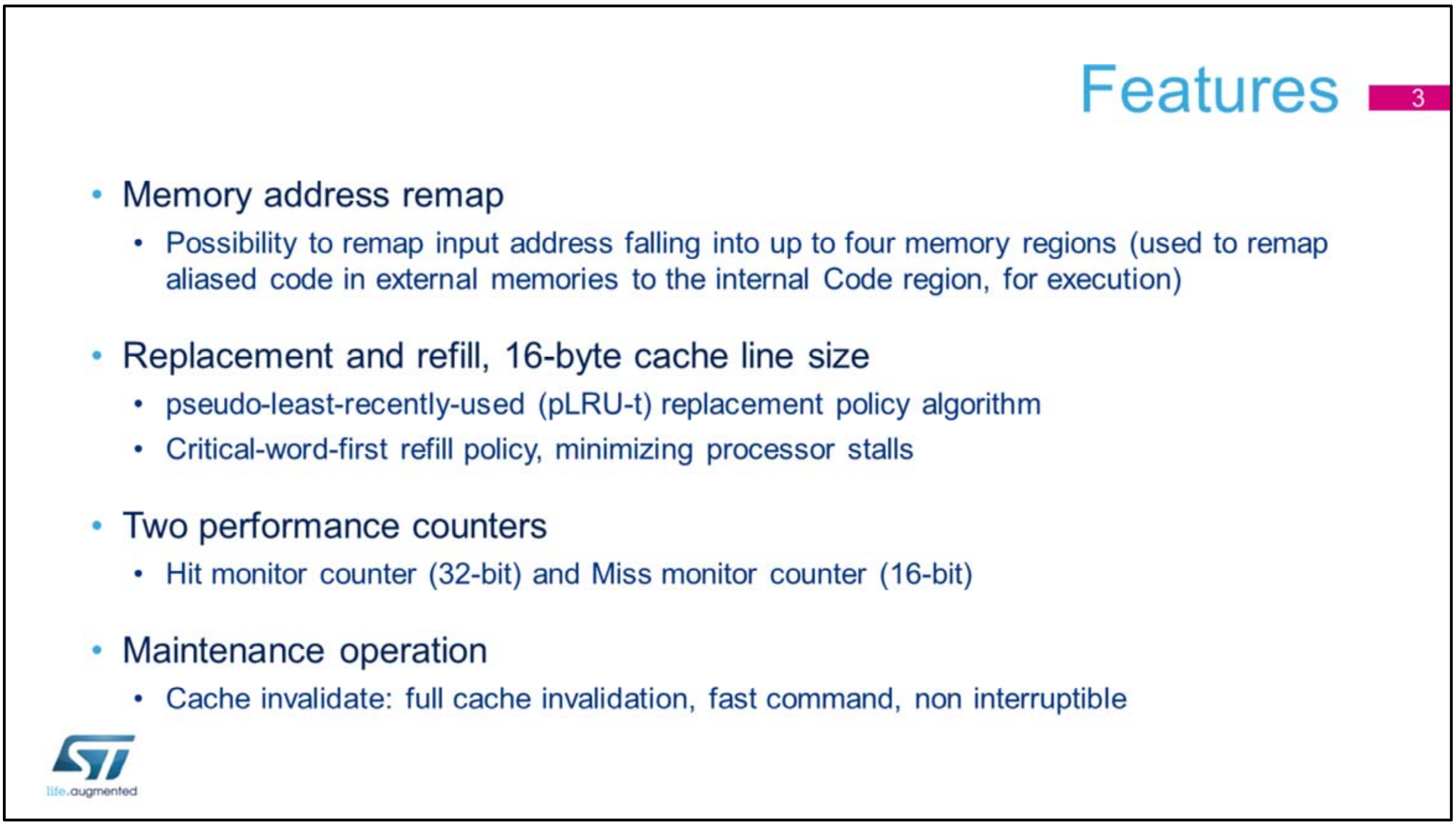
The purpose of the instruction cache is to cache instruction fetches or instruction memories loads, coming from the processor. As such ICACHE only manages read transactions and does not manage write transactions.

The ICACHE supports two configurations: 2-way set associative cache or direct-mapped.

The memory address remap capability enables accesses to external memories to be steered to the C-BUS instead of the S-BUS.

The ICACHE contributes to reducing power consumption: accessing the small internal ICACHE memory in case of a cache hit consumes less than reading from flash memory or external memories.

A software configuration of ICACHE as direct mapped allows even lower power consumption.



Features

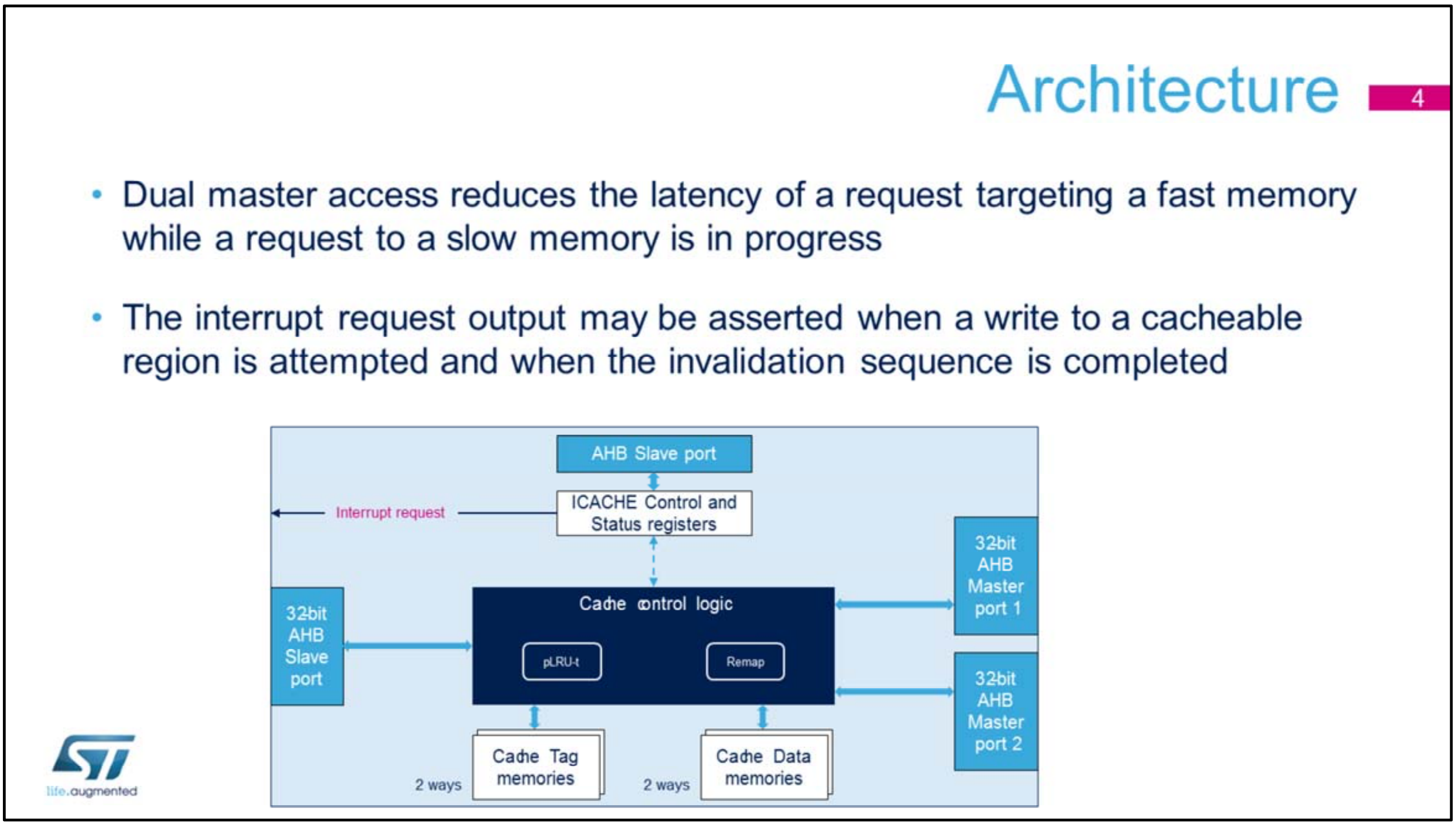
- Memory address remap
  - Possibility to remap input address falling into up to four memory regions (used to remap aliased code in external memories to the internal Code region, for execution)
- Replacement and refill, 16-byte cache line size
  - pseudo-least-recently-used (pLRU-t) replacement policy algorithm
  - Critical-word-first refill policy, minimizing processor stalls
- Two performance counters
  - Hit monitor counter (32-bit) and Miss monitor counter (16-bit)
- Maintenance operation
  - Cache invalidate: full cache invalidation, fast command, non interruptible

The ICACHE supports memory address remapping for up to 4 address regions. This improves bus transaction concurrency as accesses to external memories are performed using the C-bus instead of the S-bus.

The cache line size is 16 bytes, the replacement algorithm is pseudo least recently used, based on a binary tree and critical word first burst ordering minimizes the latency of the instruction or data requested by the processor.

Two performance counters provide statistics about the utilization of the ICACHE.

A hardware sequencer, activated by software, is in charge of invalidating the entire contents of the ICACHE.



Architecture

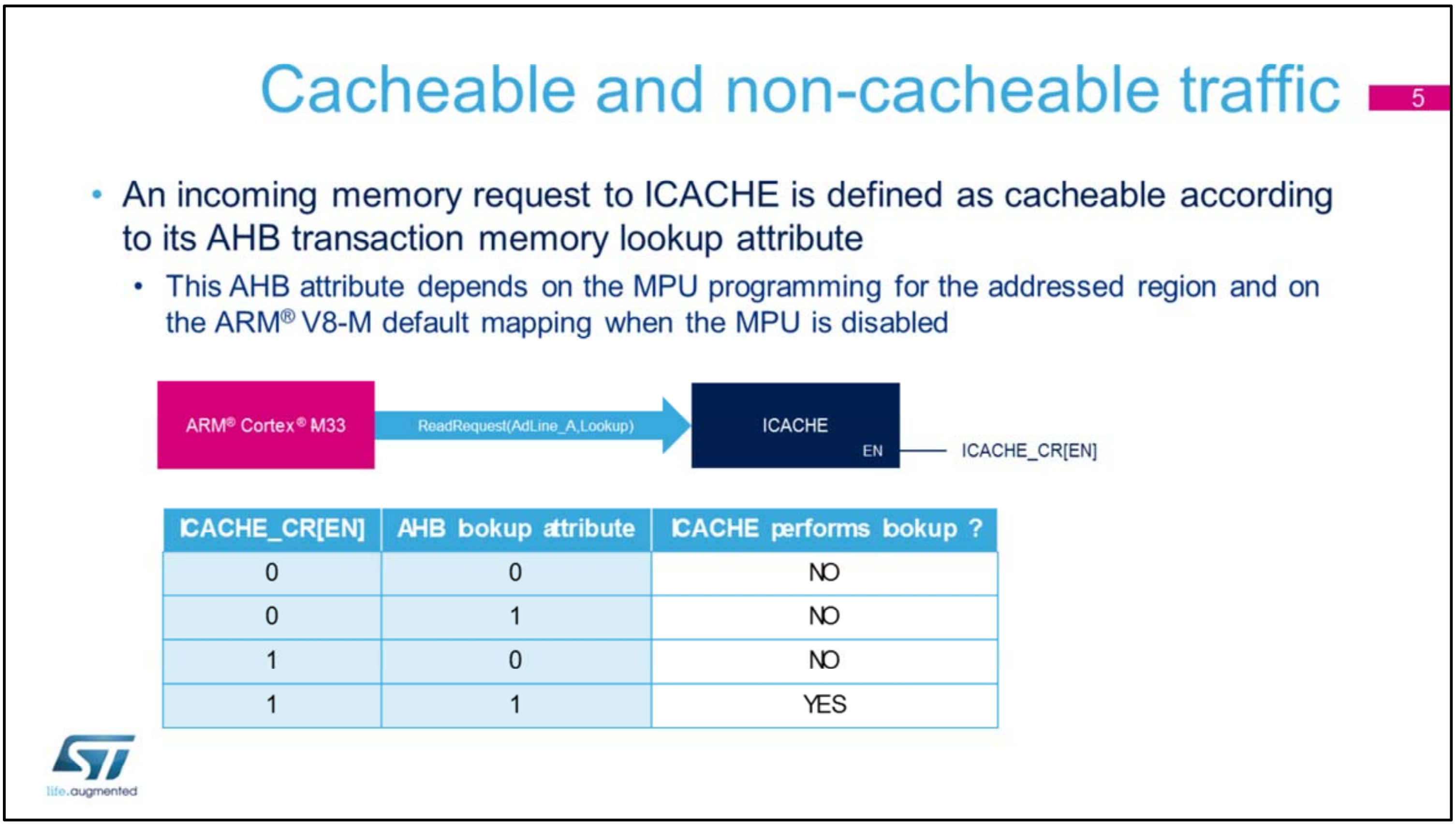
4

- Dual master access reduces the latency of a request targeting a fast memory while a request to a slow memory is in progress
- The interrupt request output may be asserted when a write to a cacheable region is attempted and when the invalidation sequence is completed

Dual master access is a feature used to decouple the traffic according to targeted memory.
For example, ICACHE assigns fast traffic (addressing FLASH and SRAM memories) to the AHB master1 port, and slow traffic (addressing external memories sitting on OCTOSPI and FMC interfaces) to the AHB master2 port, thus preventing processor stalls on line refills from external memories.
This allows ISR (interrupt service routine) fetching on internal FLASH memory to take place in parallel with a cache line refill from external memory.
The non-remapped traffic goes systematically to master1 port.
For any re-mapped region, traffic can be routed to either Master port 1 or Master port 2.
The ICACHE flags an error and possibly asserts an interrupt request whenever it detects unexpected cacheable write accesses.

An interrupt request can also be asserted upon completion of the cache invalidation sequence.

The ICACHE does not manage AHB bus errors on Master 1 or Master 2 transactions, but propagates them back to the Execution port, that received the initial C-bus transaction.



Cacheable and non-cacheable traffic

- An incoming memory request to ICACHE is defined as cacheable according to its AHB transaction memory lookup attribute
  - This AHB attribute depends on the MPU programming for the addressed region and on the ARM® V8-M default mapping when the MPU is disabled

ICACHE_CR[EN]	AHB lookup attribute	ICACHE performs lookup ?
0	0	NO
0	1	NO
1	0	NO
1	1	YES

life.augmented

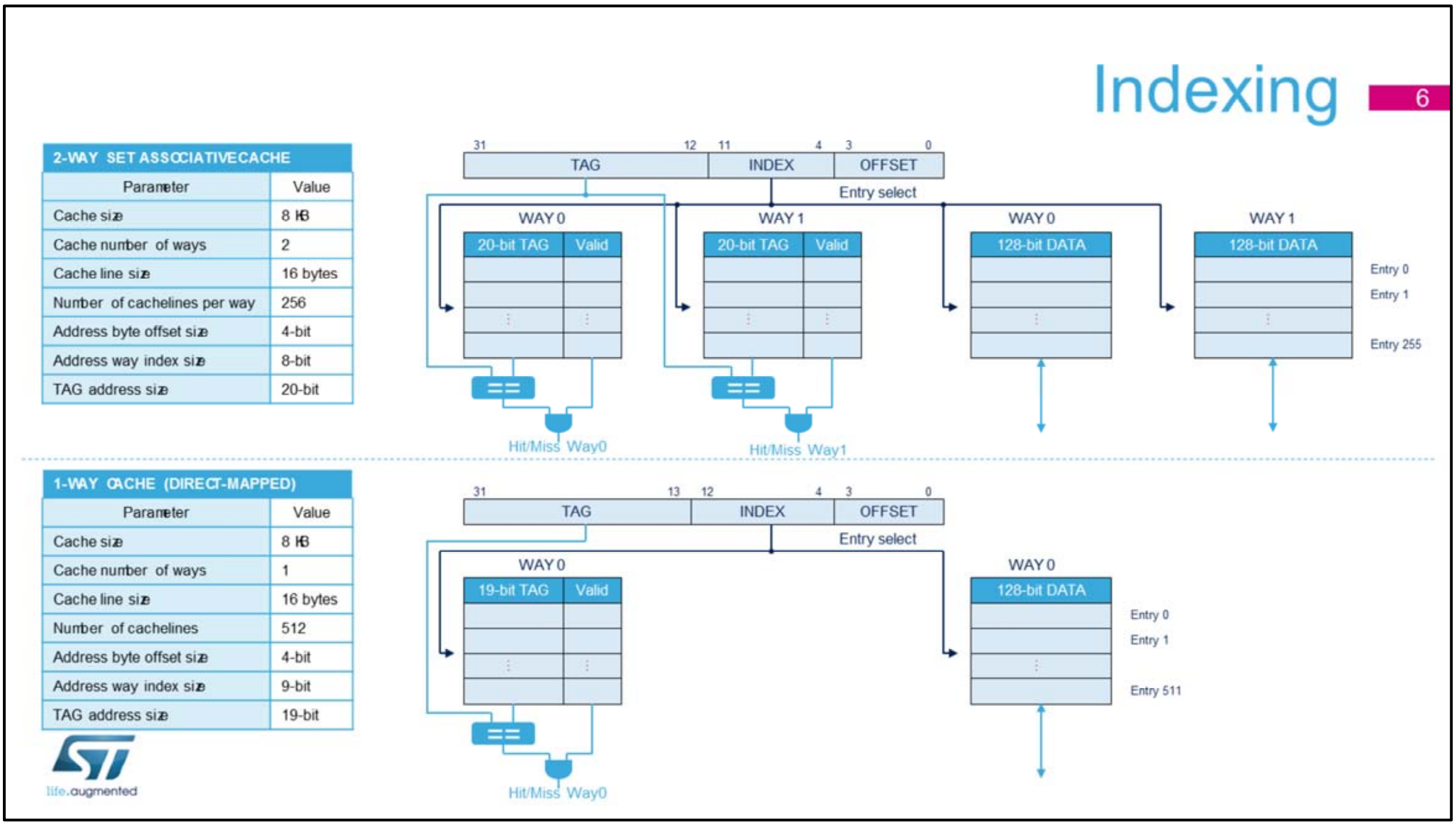
The ARM® V8-M default mapping and also the MPU define the cache attribute that the ICACHE uses to determine whether a lookup has to be performed.

Two conditions have to be satisfied to perform the cache lookup: AHB lookup attribute asserted and ICACHE enabled.

In case of a non cacheable access, ICACHE is bypassed, meaning that the AHB transaction is propagated unchanged to the master output port, except for the transaction address which may be modified due to the address remapping feature.

The bypass, and eventual remap logic, does not increase the latency of the access to the targeted memory.

ICACHE is disabled at Boot.



Indexing

6

2-WAY SET ASSOCIATIVE CACHE	
Parameter	Value
Cache size	8 KB
Cache number of ways	2
Cache line size	16 bytes
Number of cachelines per way	256
Address byte offset size	4-bit
Address way index size	8-bit
TAG address size	20-bit

1-WAY CACHE (DIRECT-MAPPED)	
Parameter	Value
Cache size	8 KB
Cache number of ways	1
Cache line size	16 bytes
Number of cachelines	512
Address byte offset size	4-bit
Address way index size	9-bit
TAG address size	19-bit

This slide details the cache organization in the two possible configurations: 2-way set associative and direct-mapped.

In 2-way set associative mode, each way contains 256 lines of 16 bytes. Thus the 4 LSbs of the address represent an offset within a cacheline and the 8-bit index selects one entry among 256 in the tag memories and in the data memories.

In direct-mapped mode, the unique way contains 512 lines of 16 bytes. The index has therefore one additional bit.

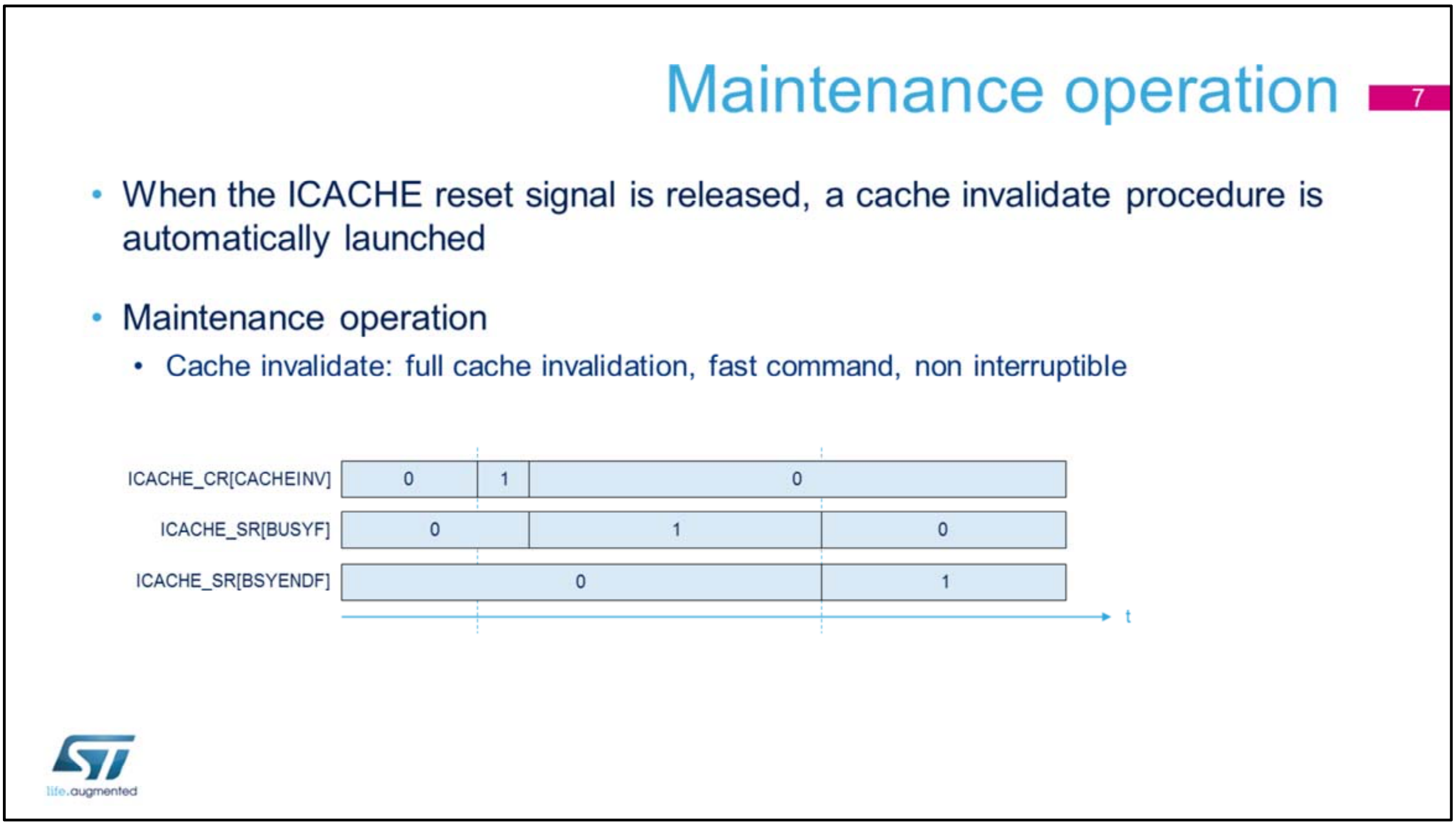
All cache operations (such as read, refill, remapping, invalidation) remain the same in direct mapped configuration; the only difference is the absence of a replacement algorithm in the case of a line eviction, since only one way (the unique one) is possible for any data refill.

The other difference is for power consumption:

in 2-way set associative mode, both cuts of memories (way-0 tags + data , and way-1 tags+data) are read speculatively at each cacheable memory request;

In direct-mapped mode, only one HW memory cut is

accessed (the one containing the 9-bit index).
So direct-mapped reduces the power consumption.



Maintenance operation

- When the ICACHE reset signal is released, a cache invalidate procedure is automatically launched
- Maintenance operation
  - Cache invalidate: full cache invalidation, fast command, non interruptible

life.augmented

A complete cache invalidation occurs in three circumstances:

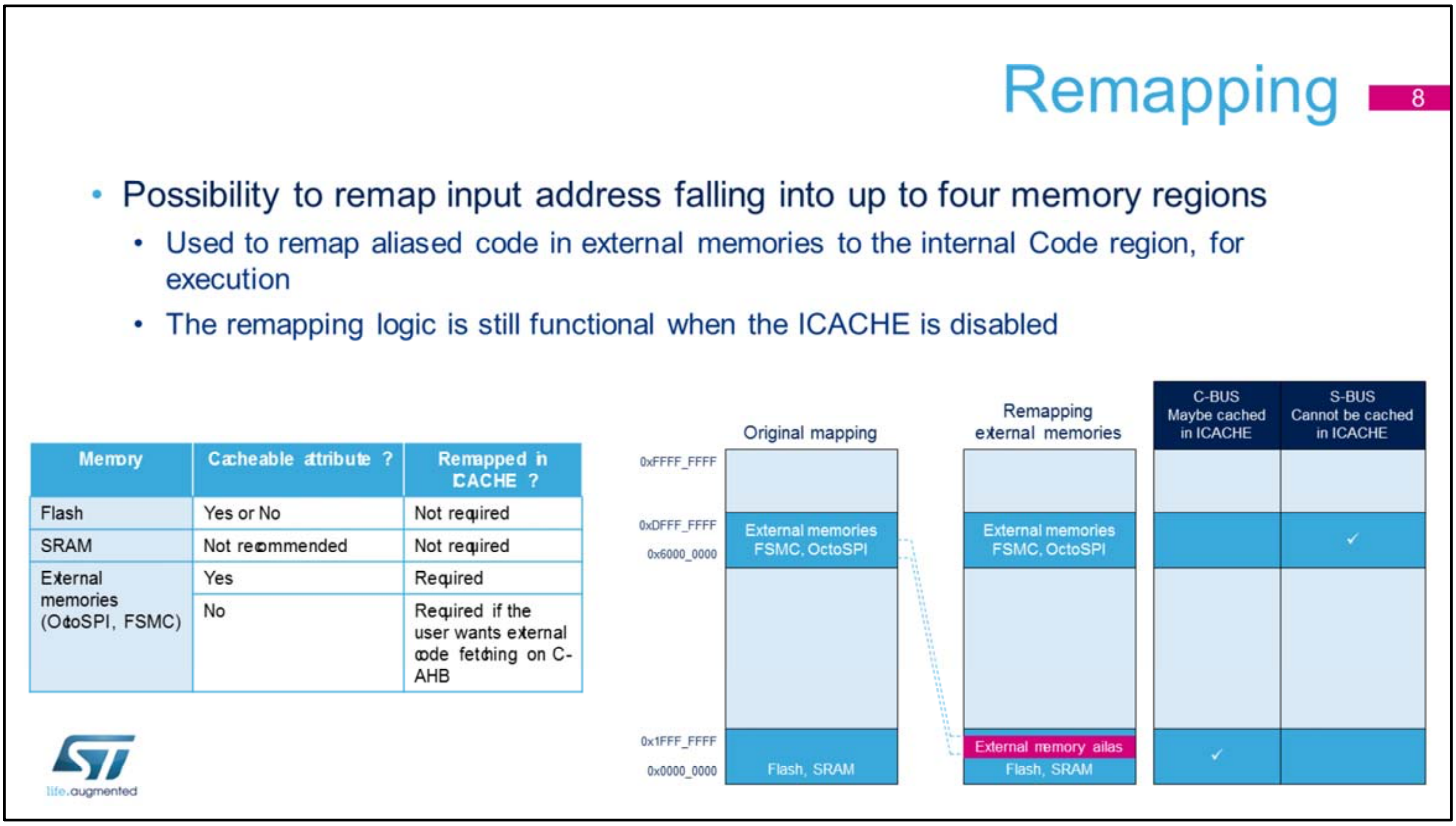
- Automatically after ICACHE reset is released
- When software sets the CACHEINV bit in the ICACHE_CR register
- When software disables the ICACHE by clearing the EN bit in the ICACHE_CR register.

Cache invalidation is performed by a hardware sequencer that sets the BUSYF bit until invalidation is completed.

The BSYENDF flag is set upon completion of the invalidation procedure and can be used to assert an interrupt request.

Software must test BUSYF and/or BSYENDF values before enabling the ICACHE.

Otherwise, if ICACHE is enabled before the end of an invalidate procedure, any cache access (while BUSYF still at 1) is treated as non cacheable.



Remapping

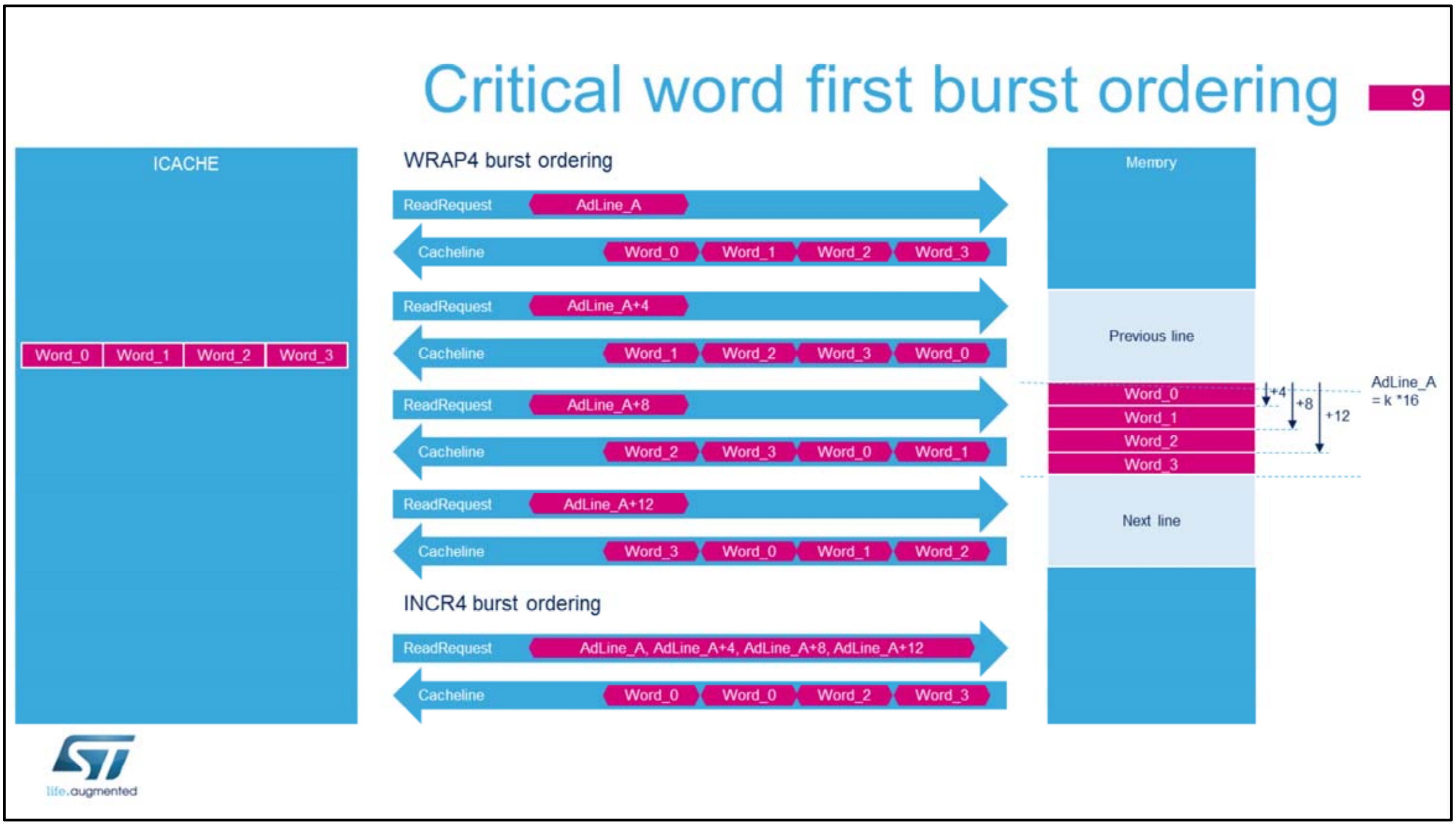
8

- Possibility to remap input address falling into up to four memory regions
  - Used to remap aliased code in external memories to the internal Code region, for execution
  - The remapping logic is still functional when the ICACHE is disabled

Memory	Cacheable attribute ?	Remapped in ICACHE ?
Flash	Yes or No	Not required
SRAM	Not recommended	Not required
External memories (OctoSPI, FSMC)	Yes	Required
	No	Required if the user wants external code fetching on C-AHB

Original mapping
Remapping external memories
C-BUS Maybe cached in ICACHE
S-BUS Cannot be cached in ICACHE
0xFFFF_FFFF
0xDFFF_FFFF
0x6000_0000
0x1FFF_FFFF
0x0000_0000
External memories FSMC, OctoSPI
External memories FSMC, OctoSPI
✓
External memory ailas
Flash, SRAM
Flash, SRAM
✓

ICACHE is placed on C-AHB bus, and thus caches the code memory region, ranging from address 0x0000 0000 to 0x1FFF FFFF of the memory map.
In order to make some other memory regions cacheable, ICACHE supports a memory region remapping feature.
Up to four external memory regions can be defined, whose addresses have an alias in the Code region. Addressing these external memory regions through their Code alias address allows the memory request to be routed to the C-AHB bus and to be managed by ICACHE.
Typically, any external memory space physically mapped at an address somewhere in the range 0x6000 0000 to 0x9FFF FFFF can be aliased with an address in range 0x0000 0000 to 0x07FF FFFF or 0x1000 0000 to 0x1FFF FFFF.
The remapping functionality is also available for non-cacheable traffic and when cache is disabled.

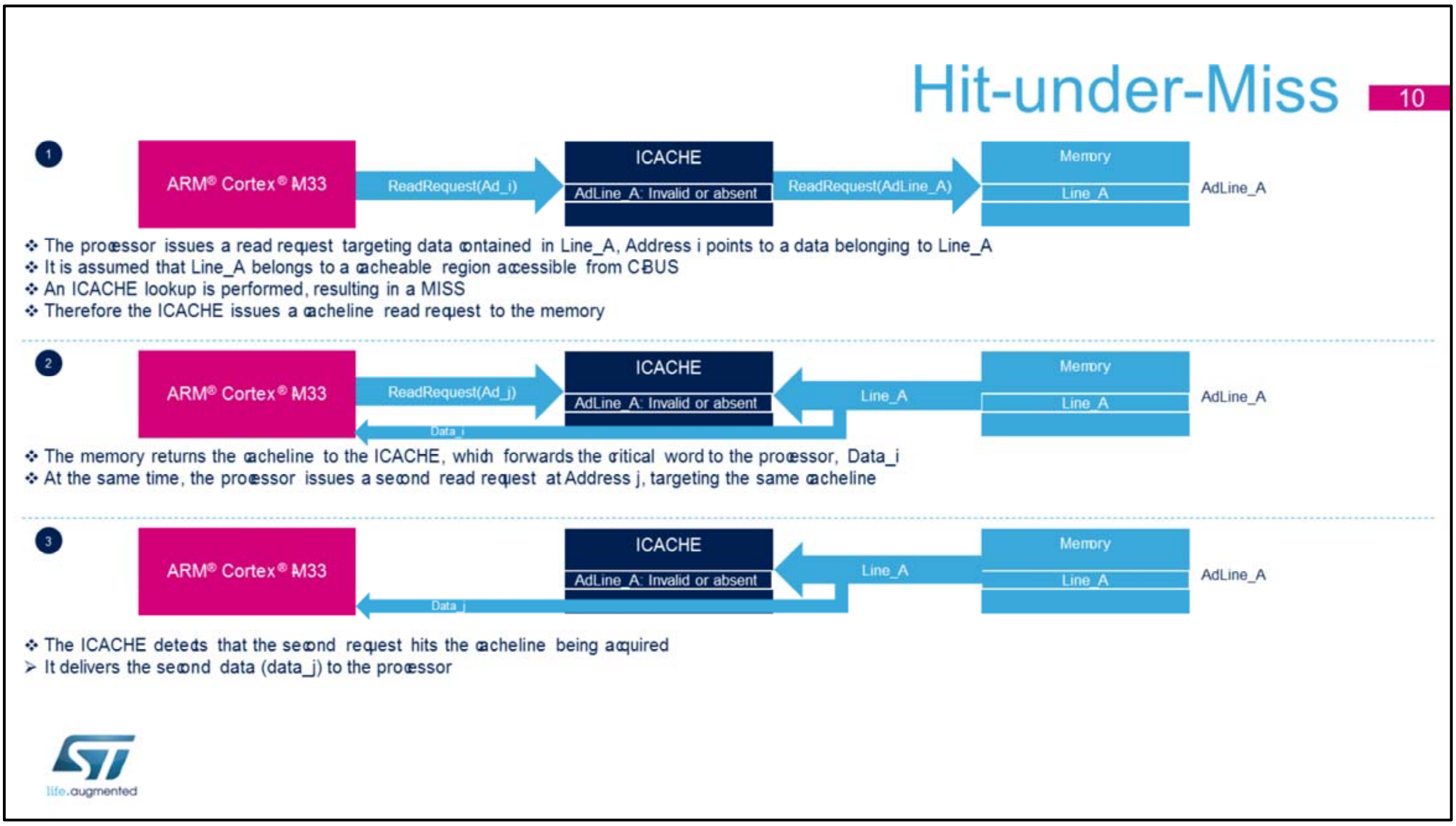


Critical word first burst ordering

The burst type of AHB memory transactions for remapped regions is programmable: INCR or WRAP.
A cacheline is aligned on its size.
WRAP4 burst ordering minimizes the latency of the instruction or data explicitly requested by the processor.
The word containing the data targeted by the address driven by the processor will be transferred first. It is called the critical word, because it contains the information actually needed by the processor. The remaining words are then transferred using the wrap ordering. For instance if the word number 3 is transferred first, there is a wrap to the beginning of the cache line and the sequence of words that follows is word 0, word 1 and word 2.
In INCR mode, it is the same word ordering (word 0, word 1, word 2 , word 3) whatever the ReadRequest address (AdLine_A, AdLine_A+4, AdLine_A+8 or AdLine_A+C).
The software can program the kind of AHB burst that is generated by ICACHE master ports, typically:

WRAP for remapped external memories accessed through the OCTOSPI interface

INCR burst mode for external memories accessed through the FSMC interface that does not support WRAP burst mode.



Hit-under-Miss

10

1

ARM® Cortex® M33

ReadRequest(Ad_i)

ICACHE

AdLine_A: Invalid or absent

ReadRequest(AdLine_A)

Memory

Line_A

AdLine_A

- The processor issues a read request targeting data contained in Line_A, Address i points to a data belonging to Line_A
- It is assumed that Line_A belongs to a cacheable region accessible from C-BUS
- An ICACHE lookup is performed, resulting in a MISS
- Therefore the ICACHE issues a cacheline read request to the memory

2

ARM® Cortex® M33

ReadRequest(Ad_j)

ICACHE

AdLine_A: Invalid or absent

Line_A

Data_i

Memory

Line_A

AdLine_A

- The memory returns the cacheline to the ICACHE, which forwards the critical word to the processor, Data_i
- At the same time, the processor issues a second read request at Address j, targeting the same cacheline

3

ARM® Cortex® M33

ICACHE

AdLine_A: Invalid or absent

Line_A

Data_j

Memory

Line_A

AdLine_A

- The ICACHE detects that the second request hits the cacheline being acquired
  - It delivers the second data (data_j) to the processor

life.augmented

The Hit-under-miss capability is the ability to serve processor requests (access to cached data) during an ongoing line refill due to a previous cache miss.

In step 1, the Cortex-M33 requests data contained in a cache line that is not currently in the cache.

A cache lookup is performed, because the region containing the target address is assumed to be cacheable, accessible from C-BUS and ICACHE is active. The result of this lookup is a cache miss.

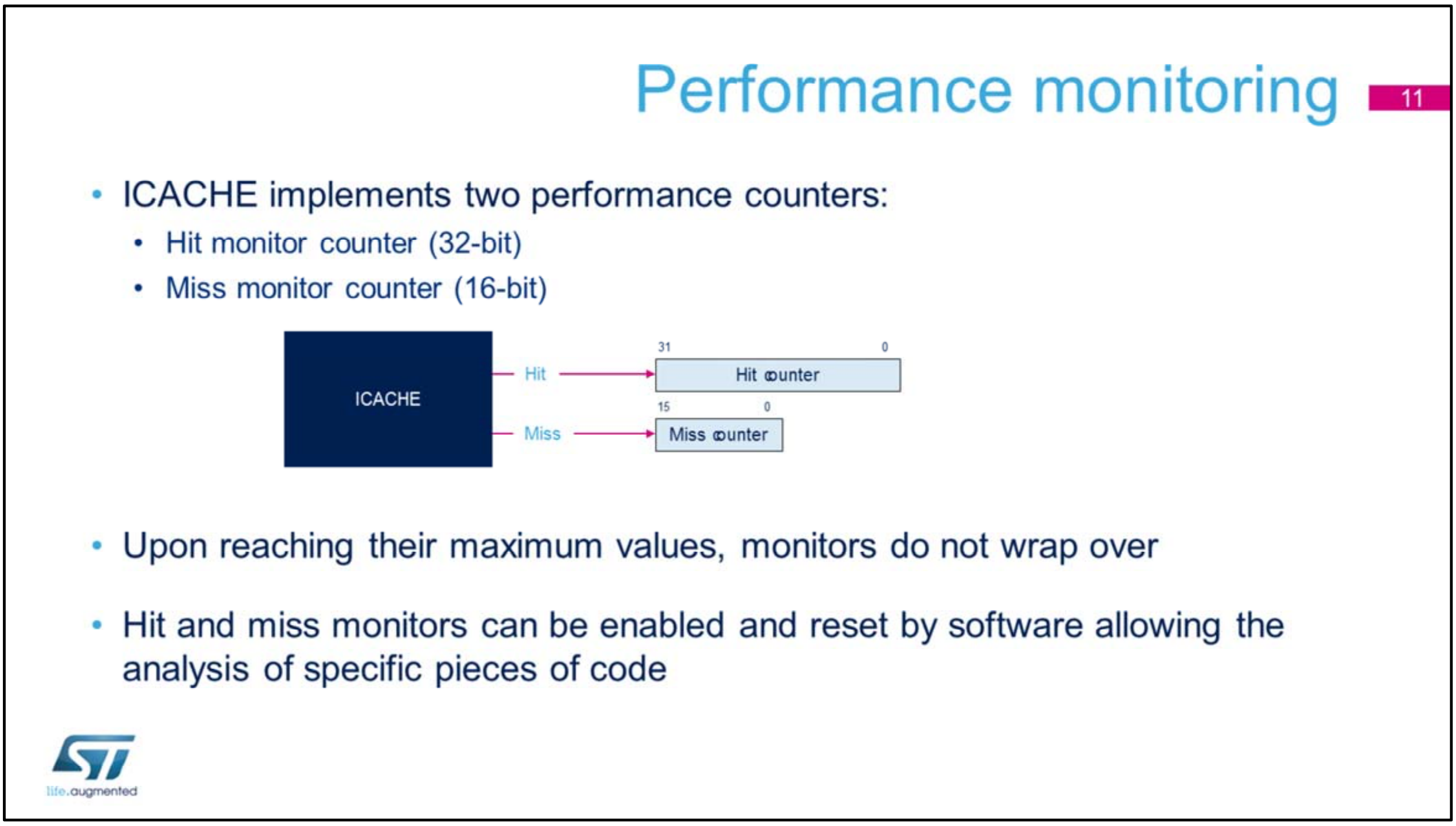
Consequently, the ICACHE issues a cache line read request to memory in order to acquire the cache line containing the data explicitly requested by the processor.

In step 2, the memory returns the cache line and forwards the critical word to the processor, data i.

At the same time, the processor issues a second request targeting the same cache line.

In Step 3, the ICACHE detects this occurrence and delivers the corresponding data to the processor, thus avoiding a cache miss.

The Hit-under-Miss feature consists in serving a hitting request just after the previous Miss, without waiting for the complete refill of the cache line.



Performance monitoring

- ICACHE implements two performance counters:
  - Hit monitor counter (32-bit)
  - Miss monitor counter (16-bit)

ICACHE
Hit
Miss
31
0
Hit counter
15
0
Miss counter

- Upon reaching their maximum values, monitors do not wrap over
- Hit and miss monitors can be enabled and reset by software allowing the analysis of specific pieces of code

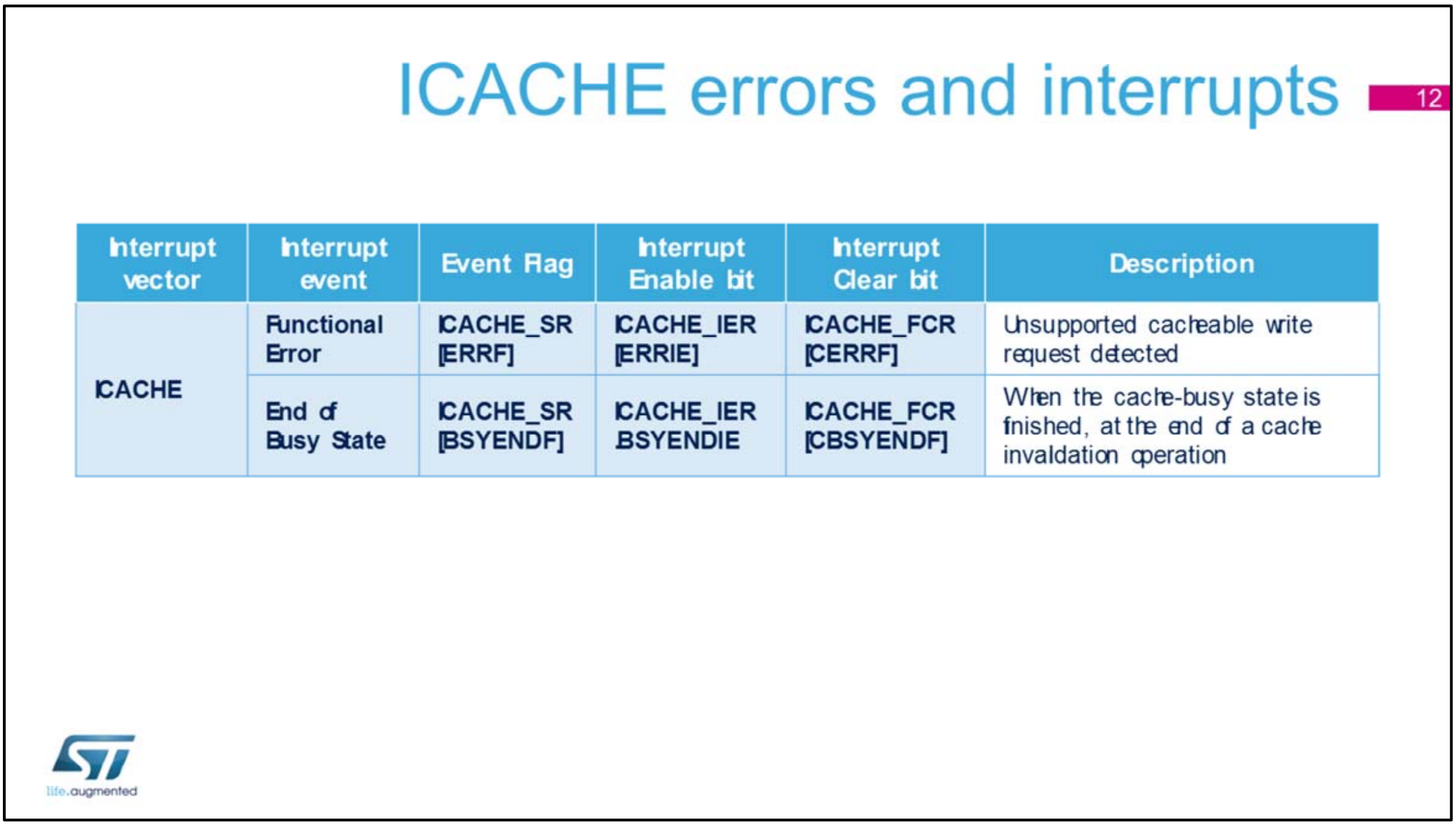
life.augmented

The hit monitor counts the AHB-transactions at the input of ICACHE (execution port) that do not generate a transaction on ICACHE output (master1 or master2 port). It also takes into account the hit under miss events.

The miss monitor counts the AHB-transactions at the input of the ICACHE (execution port) that generate a transaction on ICACHE output (master1 or master2 port). It also takes into account all accesses whose address is not present in either the TAG memory or the refill buffer.

These counters do not wrap over when they reach their maximum value.

They can be dynamically enabled and disabled by software, which is useful for analyzing specific pieces of code.



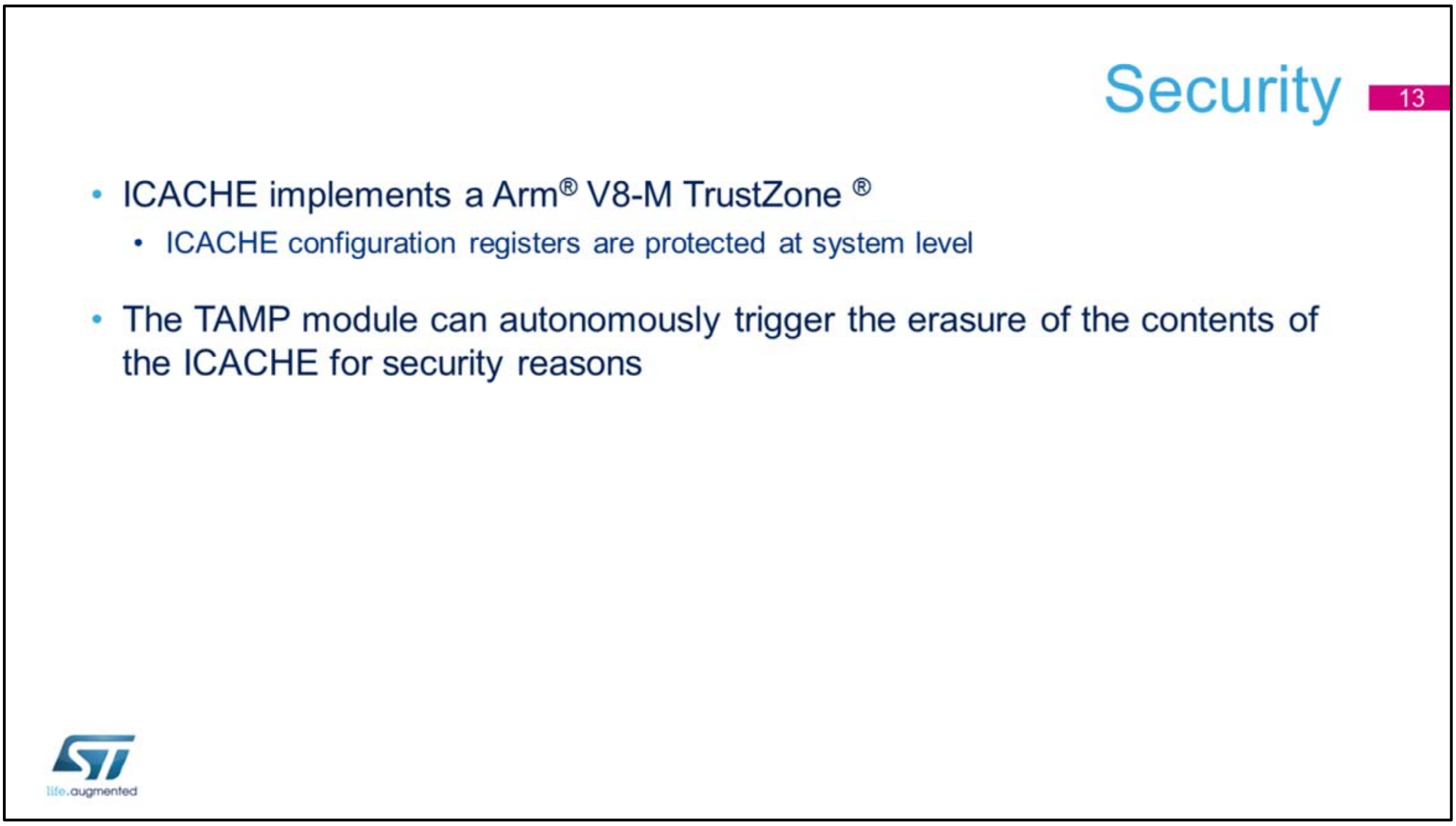
ICACHE errors and interrupts

Interrupt vector	Interrupt event	Event Flag	Interrupt Enable bit	Interrupt Clear bit	Description
ICACHE	Functional Error	ICACHE_SR [ERRF]	ICACHE_IER [ERRIE]	ICACHE_FCR [CERRF]	Unsupported cacheable write request detected
	End of Busy State	ICACHE_SR [BSYENDF]	ICACHE_IER BSYENDIE	ICACHE_FCR [CBSYENDF]	When the cache-busy state is finished, at the end of a cache invaldation operation

The ICACHE has one interrupt request output but two sources of interrupts:

- Error detection on cacheable write request (flag is SR_ERRF)
- End of Invalidate operation (flag is BSYENDF).

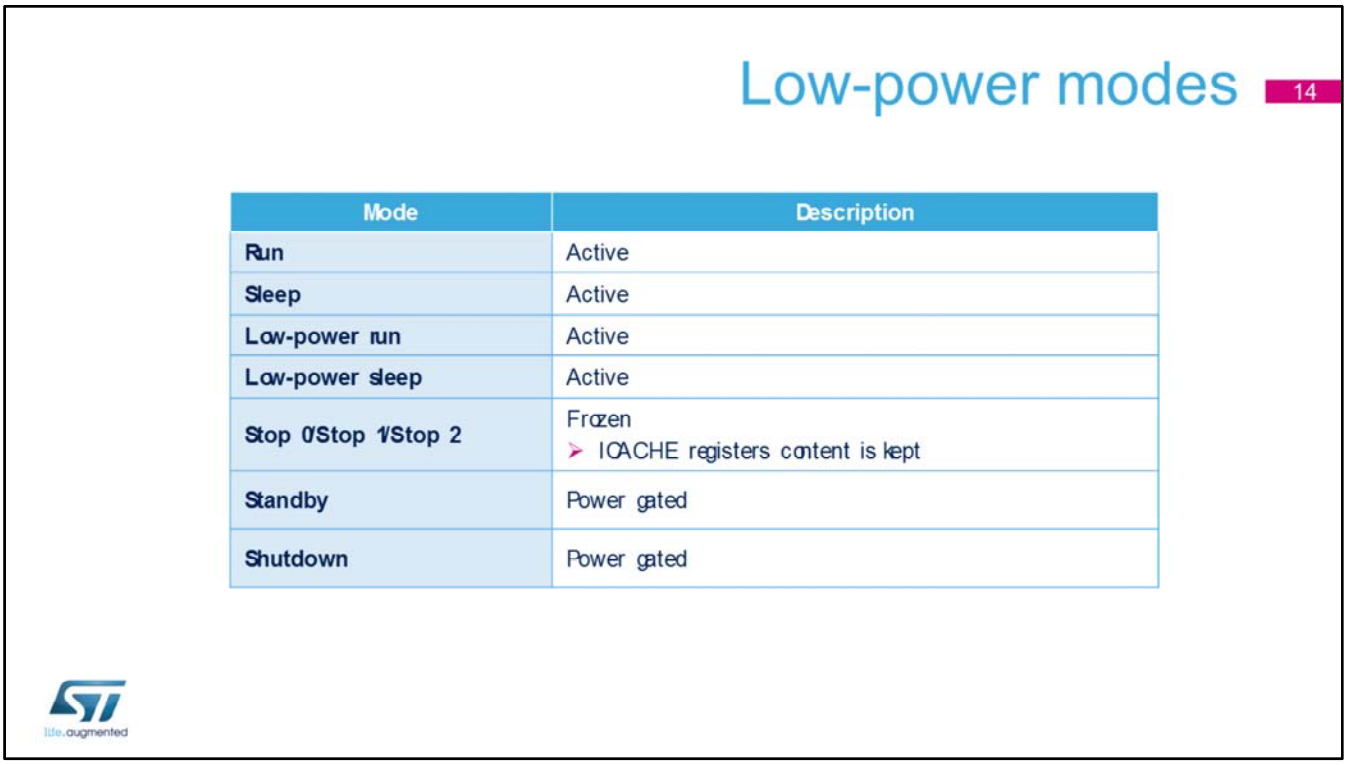
Each interrupt source has independent status, enable and clear bits.



Security

- ICACHE implements a Arm® V8-M TrustZone ®
  - ICACHE configuration registers are protected at system level
- The TAMP module can autonomously trigger the erasure of the contents of the ICACHE for security reasons

ICACHE implements a Arm® V8-M TrustZone ®.
ICACHE registers are protected at system level, enabling only secure software to access them when TrustZone is enabled.
The TAMP module can autonomously trigger the erasure of the contents of the ICACHE for security reasons.



Low-power modes

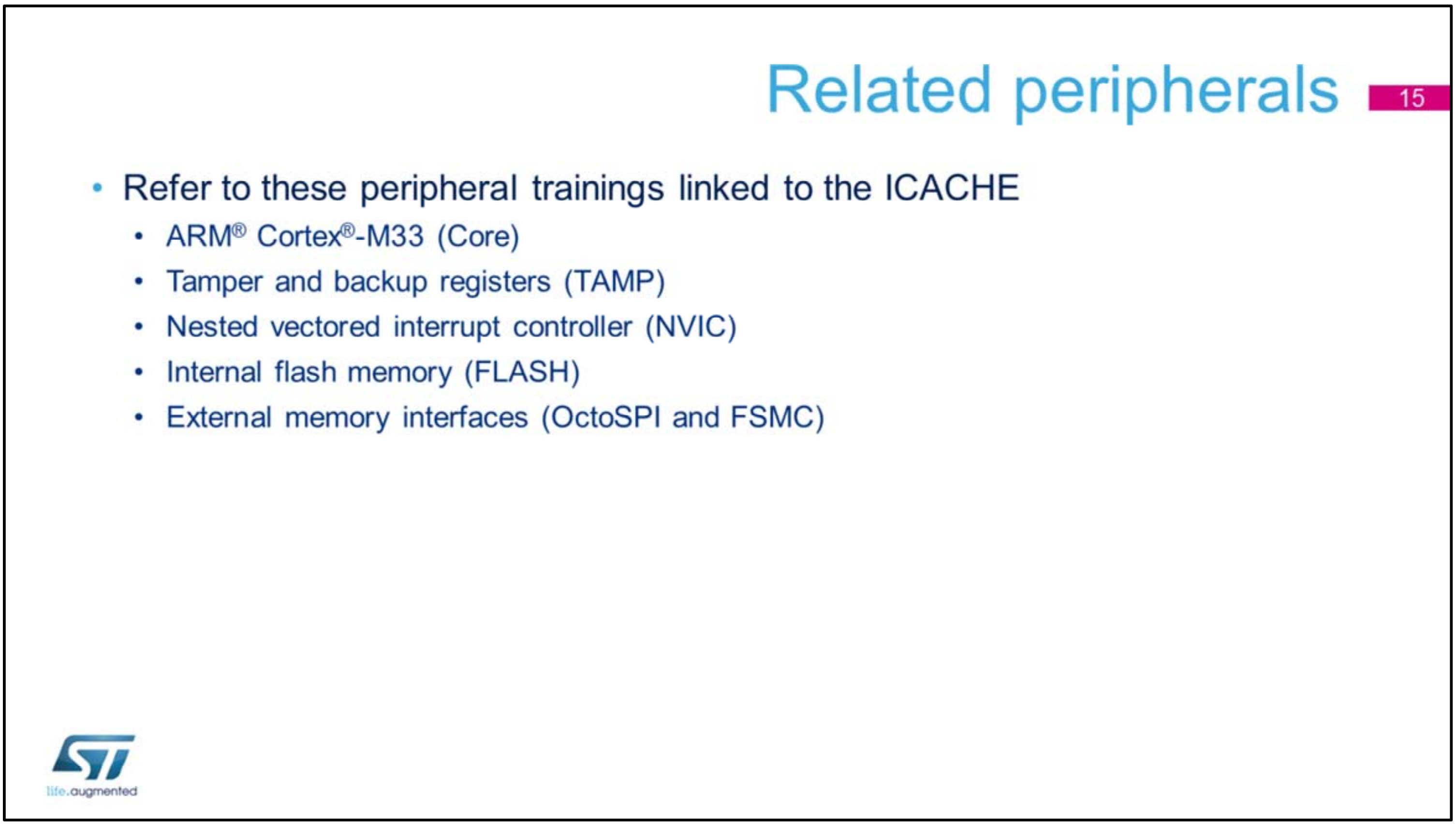
Mode	Description
Run	Active
Sleep	Active
Low-power run	Active
Low-power sleep	Active
Stop 0/Stop 1/Stop 2	Frozen ➤ ICACHE registers content is kept
Standby	Power gated
Shutdown	Power gated

ICACHE is clocked by the Cortex®-M33 C-AHB bus clock. So it has the same clock domain as the Cortex-M33 core: the same clock frequency and the same behavior during low power modes.

When disabled, ICACHE is bypassed, except for the remapping mechanism that is still functional: the Code bus input requests (remapped or not) are just forwarded to the master ports.

To reduce power consumption, the Hit and Miss monitors are disabled (stopped) by default.

They are to be used only during code debug/optimization.



Related peripherals

- Refer to these peripheral trainings linked to the ICACHE
  - ARM® Cortex®-M33 (Core)
  - Tamper and backup registers (TAMP)
  - Nested vectored interrupt controller (NVIC)
  - Internal flash memory (FLASH)
  - External memory interfaces (OctoSPI and FSMC)

This is a list of peripherals related to the ICACHE. Please refer to these peripheral trainings for more information if needed.
ARM® Cortex®-M33
Tamper and backup registers
Nested vectored interrupt controller
Internal flash memory
External memory interfaces: OctoSPI and FSMC.