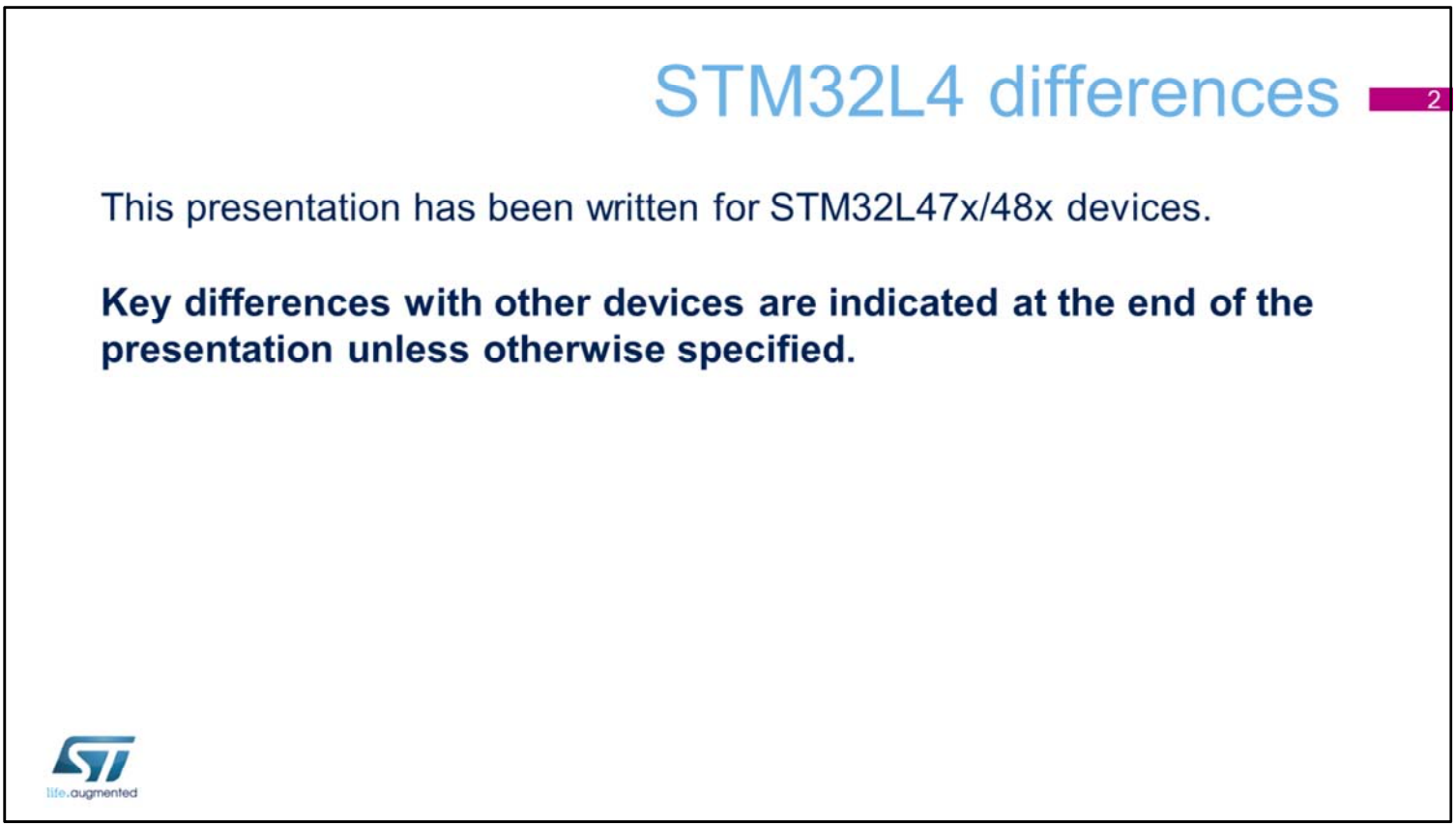
Hello, and welcome to this presentation of the STM32L4 System Configuration Controller.



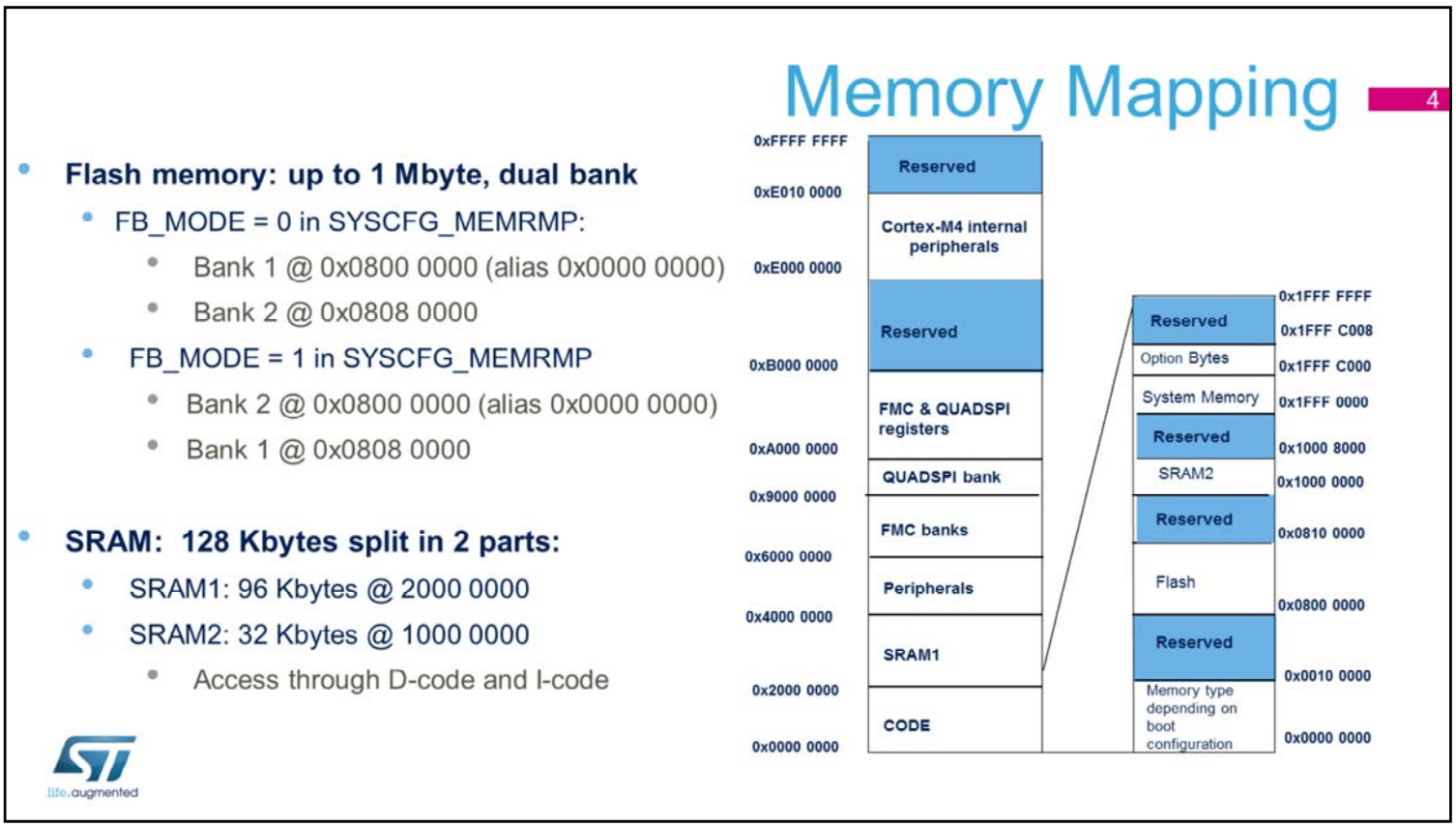
Please note that this presentation has been written for STM32L47x/48x devices.
The key differences with other devices are indicated at the end of the presentation unless otherwise specified.

Overview

- All STM32L4 devices feature a “System Configuration Controller”
  - Remap memory areas
  - Manage GPIO external interrupts
  - Manage “robustness” features
  - SRAM2 protection features
  - FPU interrupts
  - Enable the firewall
  - I2C Fast-mode Plus configuration

life.augmented

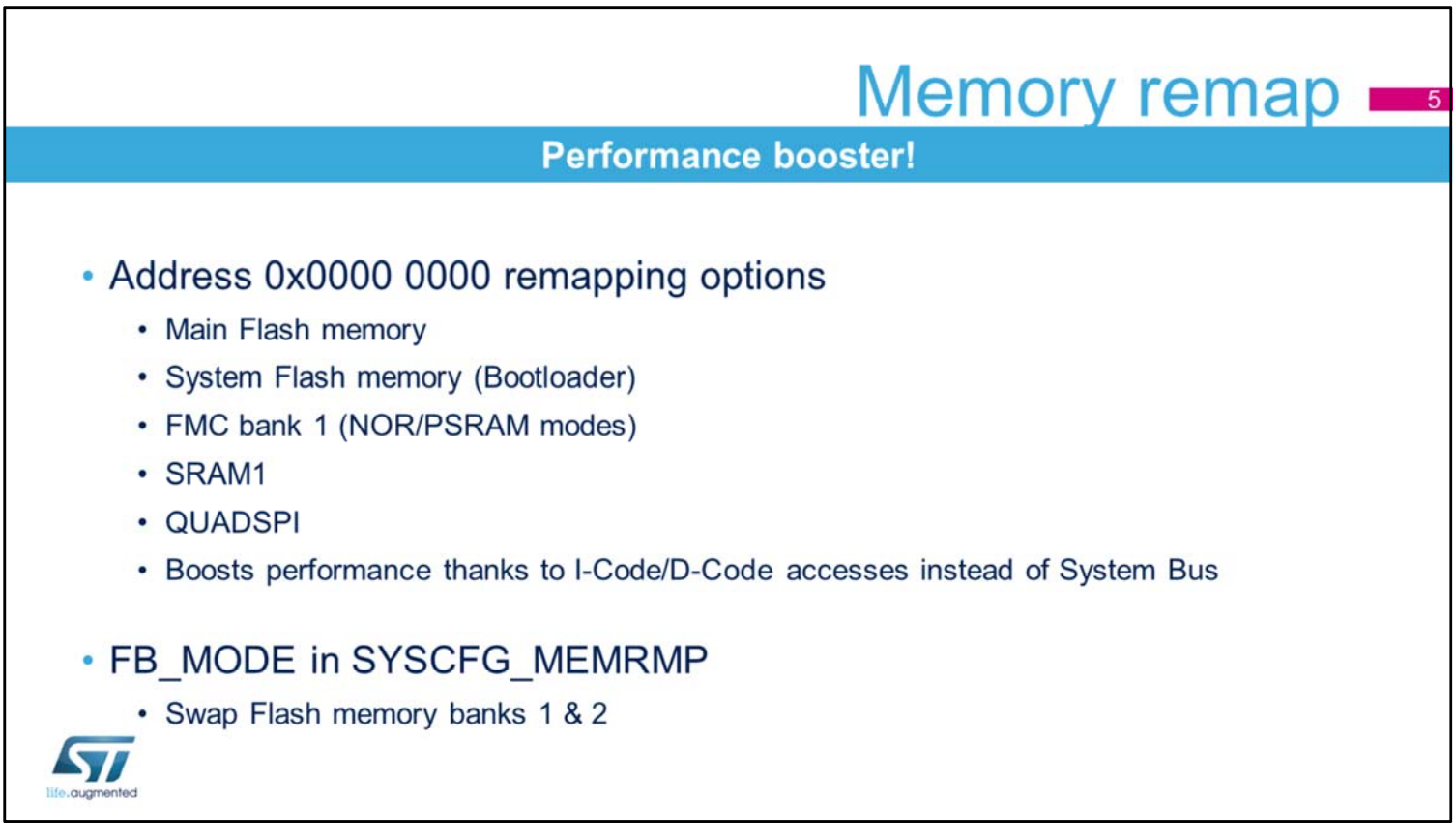
STM32L4 devices feature a set of configuration registers. The System Configuration Controller gives access to the following features: Remapping memory areas to address 0, managing the external interrupt line connection to the GPIOs, certain robustness features, SRAM2 write-protection and erase, floating point unit interrupts, firewall control and finally the configuration of the 20 mA high-drive I/Os used for I²C Fast-mode Plus.



Pictured here is the 4 gigabyte linear address mapping of the STM32L4.

The Flash memory is up to 1 Mbytes, in a dual-bank configuration. The FB_MODE bit determines the address mapping of Banks 1 and 2, as shown. It also determines which bank is aliased to address 0, which is the start of the vector table as seen by the CortexM4 core.

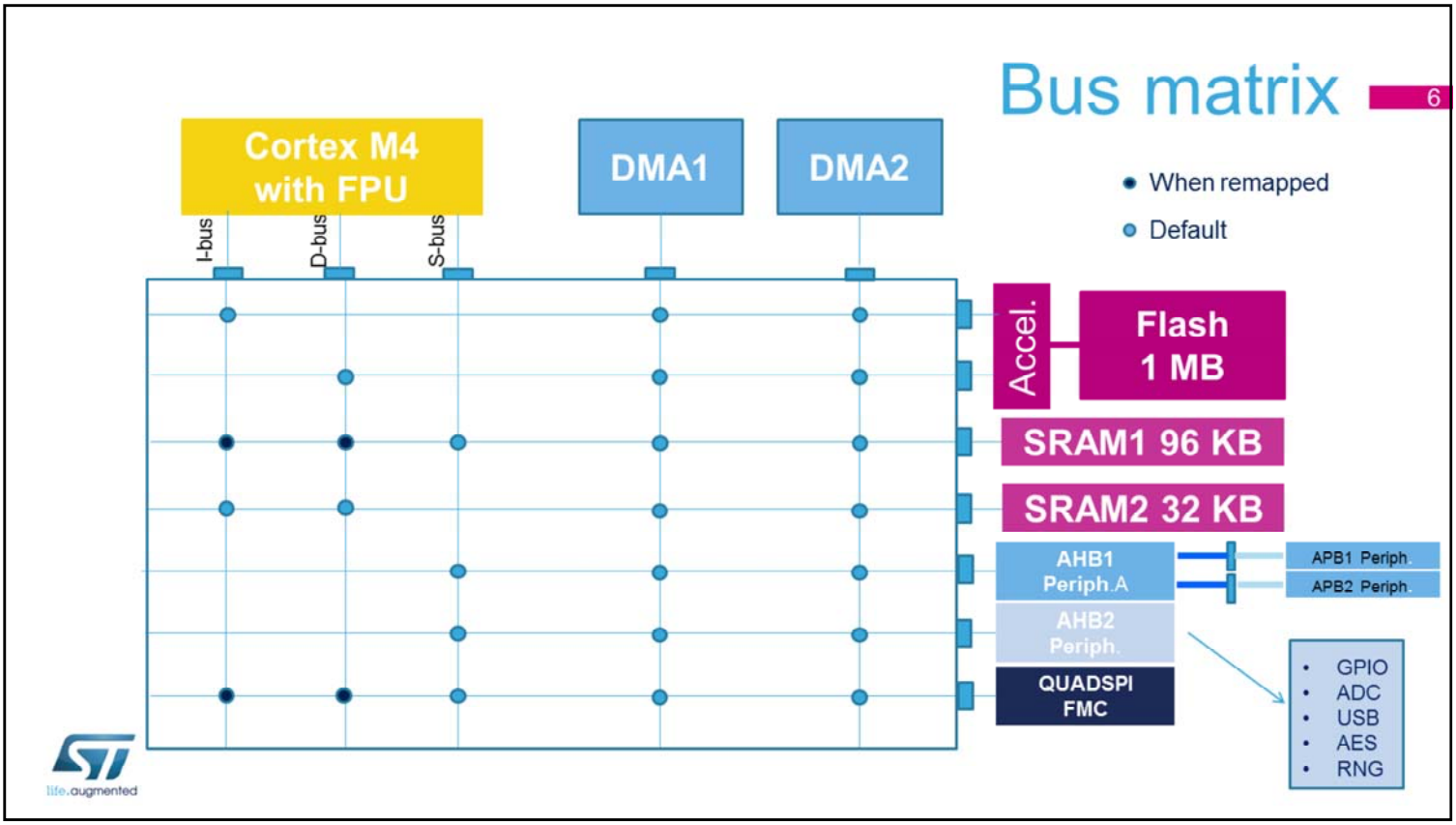
The SRAM total size is 128 Kbytes. It is split into 2 parts: SRAM1 is 96 Kbytes starting from address 0x20000000 and SRAM2 is 32 Kbytes starting from address 0x10000000. SRAM1 is located in the usual ARM memory space for RAM while SRAM2 can be directly accessed through Data code and Instruction code buses with 0 wait states, and can be used for code execution.



The memory remap at address 0 allows to boost up performance thanks to Instruction and Data bus access instead of using the System bus.

The memory remap at address 0 is selected using the MEM_MODE bits in the System Configuration Remap register. They allow to select either the main Flash memory, or the system Flash memory, the FMC bank 1 which addresses NOR or PSRAM, the SRAM1, or the QUADSPI.

The FB_MODE bit in the System Configuration Remap register allows to swap Flash memory banks 1 and 2, which allows you to boot either in Bank 1 or in Bank 2.



Here we have the STM32L4's bus matrix. The bus masters are shown on top, and the Cortex-M4 core and the two DMA controllers communicate with the bus slaves, shown on the right via the circled intersections.

The Flash memory is read through the accelerator. Cortex-M4 instructions are fetched through Instruction bus and Literal Pools are read through the Data bus. The SRAM1 is accessed by default by the System bus, and can be accessed though I-bus and D-bus when it is remapped at address 0, shown by the dark blue circles in order to increase performance. SRAM2 is always accessed through the I-bus and D-bus allowing zero-wait-state code execution. The Quad-SPI and FMC banks can be read and executed through the System bus by default, and can be remapped at 0 to increase performance. The two DMAs can access all memories

and peripherals.

Different bus masters are able to access different memories and peripherals simultaneously via the bus matrix, enabling high performance compute operations.

Boot modes

- When Boot mode = User Flash memory and option Bit BFB2 = 0
  - Boot is done in Flash memory Bank 1.
- When Boot mode = User Flash memory and option Bit BFB2 = 1
  - Boot is done in Flash memory Bank 2 if it is valid, otherwise in Bank 1.

Boot mode selection		Boot mode
nBOOT1 (option bit)	BOOT0 (pin)	
X	0	User Flash memory
1	1	System memory (bootloader)
0	1	SRAM1

There are 3 boot modes which are selected by the BOOT0 pin and an option bit named nBOOT1. When the BOOT0 pin is at a low level, the STM32L4 boots from the User Flash memory, which is aliased at address 0. This is the standard method of booting the STM32L4.

When the BOOT0 pin is at a high level, the nBOOT1 bit determines the boot mode. The default option bit setting is high, enabling the bootloader in the system memory portion of the Flash memory. The other option is booting from the SRAM1 memory region, which may be used for debugging purposes.

When the selected boot is the User Flash memory, the STM32L4 boots from Bank 1 if the option bit BFB2 is ‘0’, which is the default value. If the BFB2 option bit is set to ‘1’, the STM32L4 boots from the Flash memory Bank 2,

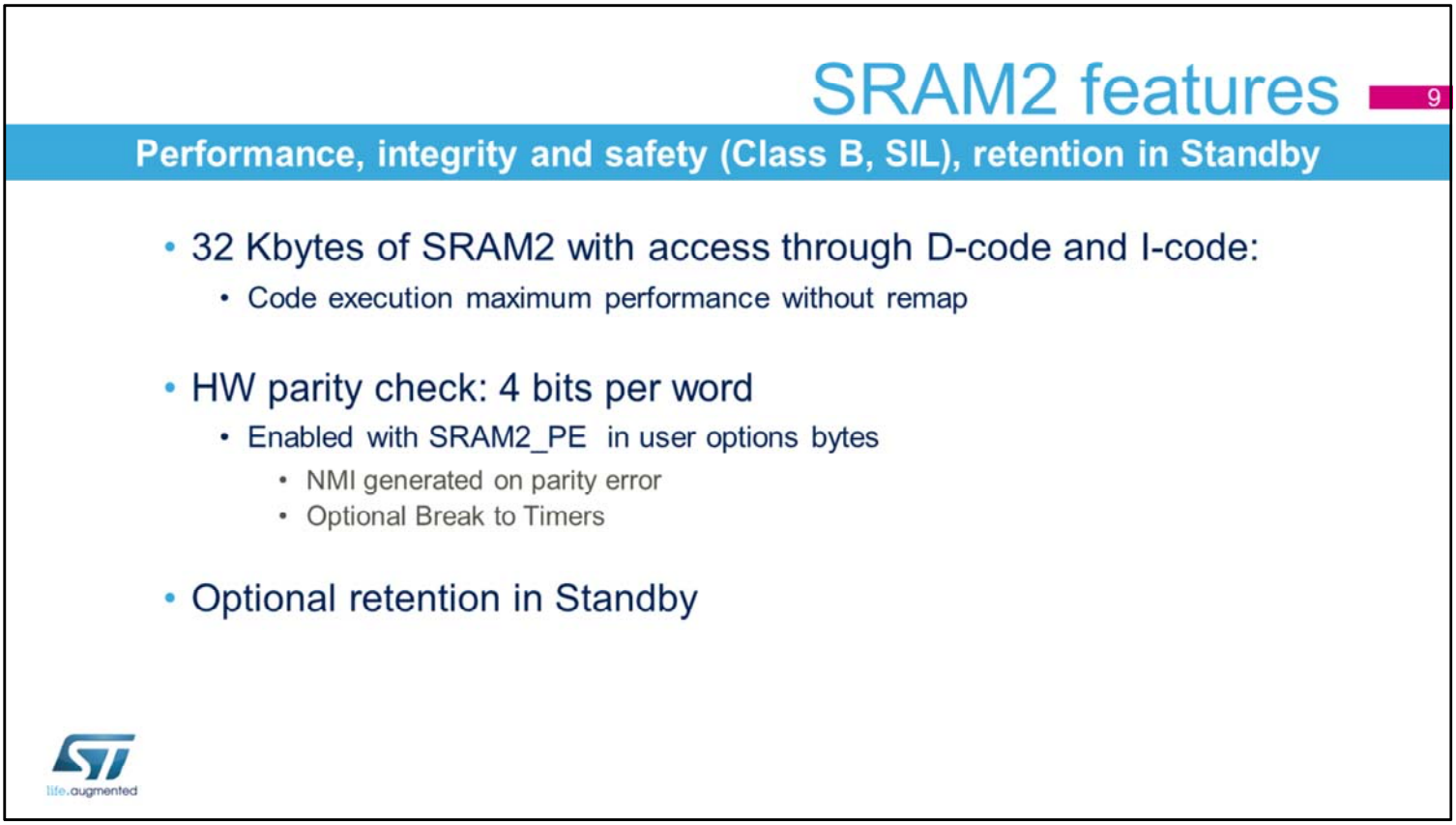
as long as its first address is a valid SRAM address. Otherwise it boots from Bank 1. This check ensures a valid vector table.

Bootloader

Protocol	I/Os and Comments	Comments
USART	USART1 on pins PA9/PA10 USART2 on pins PA2/PA3 USART3 on pins PC10/PC11	
USB	USB DFU interface on pins PA11/PA12	Bootloader checks if HSE present : USB clock is HSE If no Bootloader checks if LSE present : USB clock is MSI auto-trimmed with LSE
CAN	CAN1 on pins PB8/PB9	
SPI	SPI1 on pins PA4/PA5/PA6/PA7 SPI2 on pins PB12/PB13/PB14/PB15	
I2C	I2C1 on pins PB6/PB7 I2C2 on pins PB10/PB11 I2C3 on pins PC0/PC1	I²C slave address is 0x86

life.augmented

The on-chip bootloader allows the user to program the Flash memory through a serial communications peripheral. The supported protocols are USART, USB, CAN, SPI and I²C.



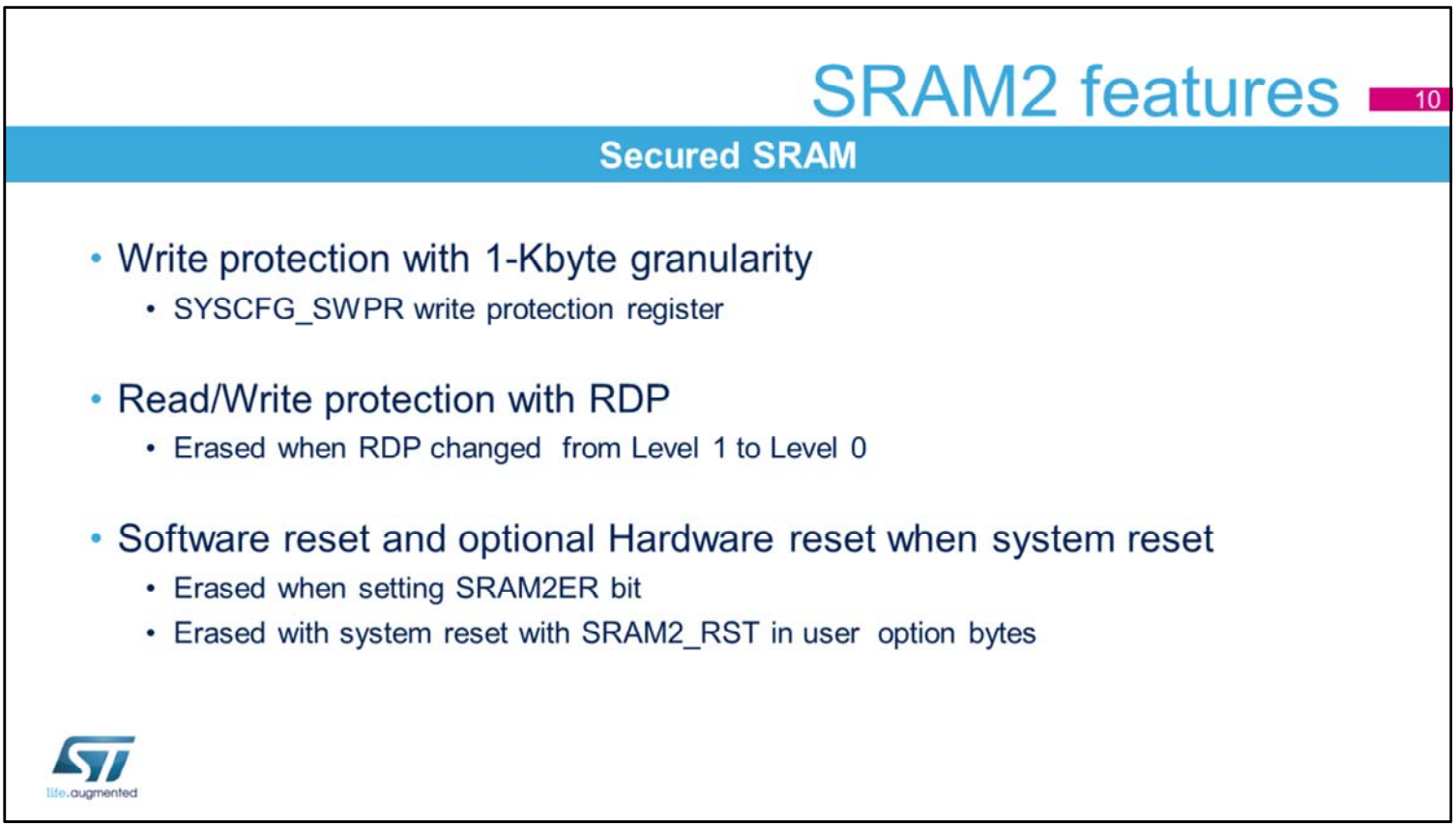
The 32 Kbytes of SRAM2 is particularly suitable for performance, integrity and safety, and low power.

The SRAM2 is accessed through the Data and Instruction buses without any remapping, which enables code execution at zero-wait-states.

The SRAM2 supports parity check. The Data bus width is 36 bits because 4 bits are available for parity check (1 bit per byte) in order to increase memory robustness, as required, for instance, by Class B or SIL standards. Class B and SIL are safety standards: Class B is for Home Appliances and SIL for the Safety Integrity Level. The parity bits are computed and stored when writing into the SRAM. Then, they are automatically checked when reading. If one bit fails, an NMI is generated. The same error can also be linked to the Break input of the

timers. Note that the SRAM2 parity check is disabled by default.

The SRAM2 content can optionally be retained in Standby.



The SRAM2 is also suitable for secure applications. The SRAM2 can be write-protected with a 1-Kbyte granularity.

The SRAM2 can also be readout-protected via the RDP option byte. When protected, the SRAM2 cannot be read or written by the JTAG or serial wire debug port, and when the boot in System flash or boot in SRAM is selected. The SRAM2 is erased when the readout protection is changed from Level 1 to Level 0. Please refer to the System Memory Protections training for further details.

The SRAM2 can be erased by software by setting the SRAM2ER bit in the SRAM2 System Configuration Control and Status register. The SRAM2 can also be erased with the system reset depending on the option bit

SRAM2_RST in the user option bytes.

SYSCFG other features

Safety and robustness

- Safety & Robustness features in Configuration register 2
  - SRAM2 Parity error flag
  - ECC lock to connect Flash ECC error connection to TIM1/8/15/16/17 Break input
  - PVD lock to connect PVD interrupt to TIM1/8/15/16/17 Break input
  - SRAM2 parity error connection to TIM1/8/15/16/17 Break input
  - CLL lock to connect Cortex M4 Hard Fault interrupt to TIM1/8/15/16/17 Break input
  - => Put timers in application safe state in case of application crash

life.augmented

The System Configuration Register 2 contains the control and status bits linked to safety and robustness such as the SRAM2 parity error flag, and the control bits to direct some error detections events to the timers' break inputs. This allows timer outputs to be placed in a known state during an application crash. Once programmed, the connection is locked until the next system reset. These internal events include a Flash error-code-correction event, a power voltage detector event, SRAM2 parity error event, and the Cortex M4 hard fault.

SYSCFG other features

12

- Manage external interrupt (EXTI) connection to GPIOx (x=A,…H)
  - 16 Multiplexers to select EXTIn between PA[n] PB[n] …. PH[n] (n=0,…15)
- Configuration register 1
  - FPU interrupts enable
  - I2C GPIO Fast-mode Plus 20 mA drive enable
    - PB6, PB7, PB8, PB9 high drive can be enabled even when not used for I2C
  - I/O analog switches voltage booster
  - Firewall enable

life.augmented

The System Configuration Controller manages the selection of the GPIO to the external interrupt or event signal, which is used as asynchronous external interrupt or event with wakeup from Stop capability.

Configuration register 1 contains the floating point unit interrupt control bits. It contains also the I²C Fast-mode-Plus 20 mA drive enable control bits. Four I/Os can be configured with high drive mode even if they are not used as I2C alternate functions. They can be used to drive LEDs for instance.
The I/O analog switch voltage booster is also selected here as well as the Firewall.

Performance

	Execution in Flash memory			Execution in SRAM	
	ART ON I-Cache ON D-Cache ON Prefetch ON	ART ON I-Cache ON D-Cache ON Prefetch OFF	ART OFF	Code & Data in SRAM1	Code in SRAM2, Data in SRAM1
CoreMark / MHz @ 80 MHz	3.35	3.32	1.55	2.37	3.42

Here we compare code execution performance at 80 MHz while running the EEMBC CoreMark benchmark.

The maximum performance is reached when the code is executed in SRAM2 with data located in SRAM1. It is also possible to reach maximum performance with code in SRAM1 and data in SRAM2 if the SRAM1 is remapped at address 0.
When executing from Flash memory at 80 MHz, the maximum CoreMark performance is reached when the ART accelerator is enabled, and there is almost no loss of performance due to the Flash access time requiring 4 wait states at 80 MHz. Enabling the prefetch buffer yields a slightly higher score, 3.35 CoreMark / MHz.

Related peripherals

- Refer to these training modules linked to this peripheral:
  - Reset and clock control (RCC)
  - Power controller (PWR)
  - Interrupts (NVIC-EXTI)
  - Flash memory (Flash)
  - System memory protections
  - Timers (TIM)
  - Inter-Integrated Circuit (I²C)

In addition to this training, you can refer to the Reset and Clock Control, Power Controller, Interrupts, Flash and System Memory Protections, Timers and I²C trainings.

References

- For more details, please refer to following resources:
  - AN2606: STM32 microcontroller system memory boot mode
  - AN4435: Guidelines for obtaining UL/CSA/IEC 60335 Class B certification in any STM32 application

life.augmented

For more details, please refer to application notes AN2606 STM32 microcontroller system memory boot mode and AN4435 Guidelines for obtaining UL/CSA/IEC 60335 Class B certification in any STM32 application.

Differences with STM32L47x/48x devices (1)

- Multilayer AHB matrix
  - STM32L49x/4A6: additional DMA2D Chrome ART accelerator master split of FMC/QUADSPI into separate slaves.
  - STM32L41x/42x/43x/44x/45x/46x: no FMC slave.
- SRAM size
  - STM32L49x/4A6: SRAM1: 256 KB SRAM2: 64 KB
  - STM32L47x/48x: SRAM1: 96 KB SRAM2: 32 KB
  - STM32L45x/46x: SRAM1: 128 KB SRAM2: 32 KB
  - STM32L43x/44x: SRAM1: 48 KB SRAM2: 16 KB
  - STM32L41x/42x: SRAM1: 32 KB SRAM2: 8 KB
- Memory mapping:
  - STM32L41x/42x/43x/44x/45x/46x/49x/4Ax: SRAM2 is aliased at @0x2000 0000 + SRAM1 size to allow continuous address space with SRAM1.
- FLASH :
  - STM32L41x/42x/43x/44x/45x/46x: Single bank FLASH. No FB_MODE

life.augmented

This slide presents the key differences between baseline STM32L47x/48x devices and other devices.
Multilayer AHB matrix is bigger for STM32L49x/4Ax devices, including new master DMA2D Chrome Art accelerator for graphic and splitting FMC/QUADSPI slave into 2 separated slaves in order to increase bandwidth in case of FMC working in parallel to QUADSPI.
On STM32L41x/42x/43x/44x/45x/46x devices there is no FMC slave and the flash memory is single bank.
The SRAM sizes are different for each device and are listed here.
On STM32L4 derivatives the SRAM2 is aliased to offer a continuous address space with SRAM1.

Differences with STM32L47x/48x devices (2)

- Boot mode:
  - On STM32L41x/42x/43x/44x/45x/46x devices, the Boot0 pin is shared with PH3 GPIO.
  - Two additional option bytes (nBOOT0 and nSWBOOT0) are used to select the boot mode.
  - Also, a Flash Empty Check mechanism is implemented (*).

Boot mode selection					Boot mode
nBOOT1 (option byte)	nBOOT0 (option byte)	BOOT0/PH3 (pin)	nSWBOOT0 (option byte)	Main Flash empty	
x	x	0	1	0	User Flash memory
x	x	0	1	1	System memory
x	1	x	0	x	User Flash memory
0	x	1	1	x	SRAM1
0	0	x	0	x	SRAM1
1	x	1	1	x	System memory
1	0	x	0	x	System memory

life.augmented

(*) To force the boot from System Flash instead of the main Flash if the first Flash memory location is not programmed.

On STM32L4 derivatives, the boot mode is selected either with the nBOOT1 option bit and pin BOOT0 or with the nBOOT0 option bit, depending on the value of the nSWBOOT0 option bit in the FLASH_OPTR register as shown in the table.
A Flash Empty Check mechanism is implemented to force the boot from System memory instead of the main Flash memory if the first Flash memory location is not programmed.
This feature is only available on derivative products of STM32L4 series.

Differences with STM32L47x/48x devices (3)

- Boot loader:

  On STM32L45x/46x/49x/4A6 devices it is possible to boot from additional peripheral I2C4
  On STM32L49x/4A6 devices it is possible to boot from additional peripheral CAN2

Protocol	I/Os and Comments	Comments
CAN	CAN2 on pins PB5/PB6	
I2C	I2C4 on pins PD12/PD13	I²C slave address is 0x86

life.augmented

On STM32L4 derivatives, it is possible to boot from additional peripherals as listed in this slide.