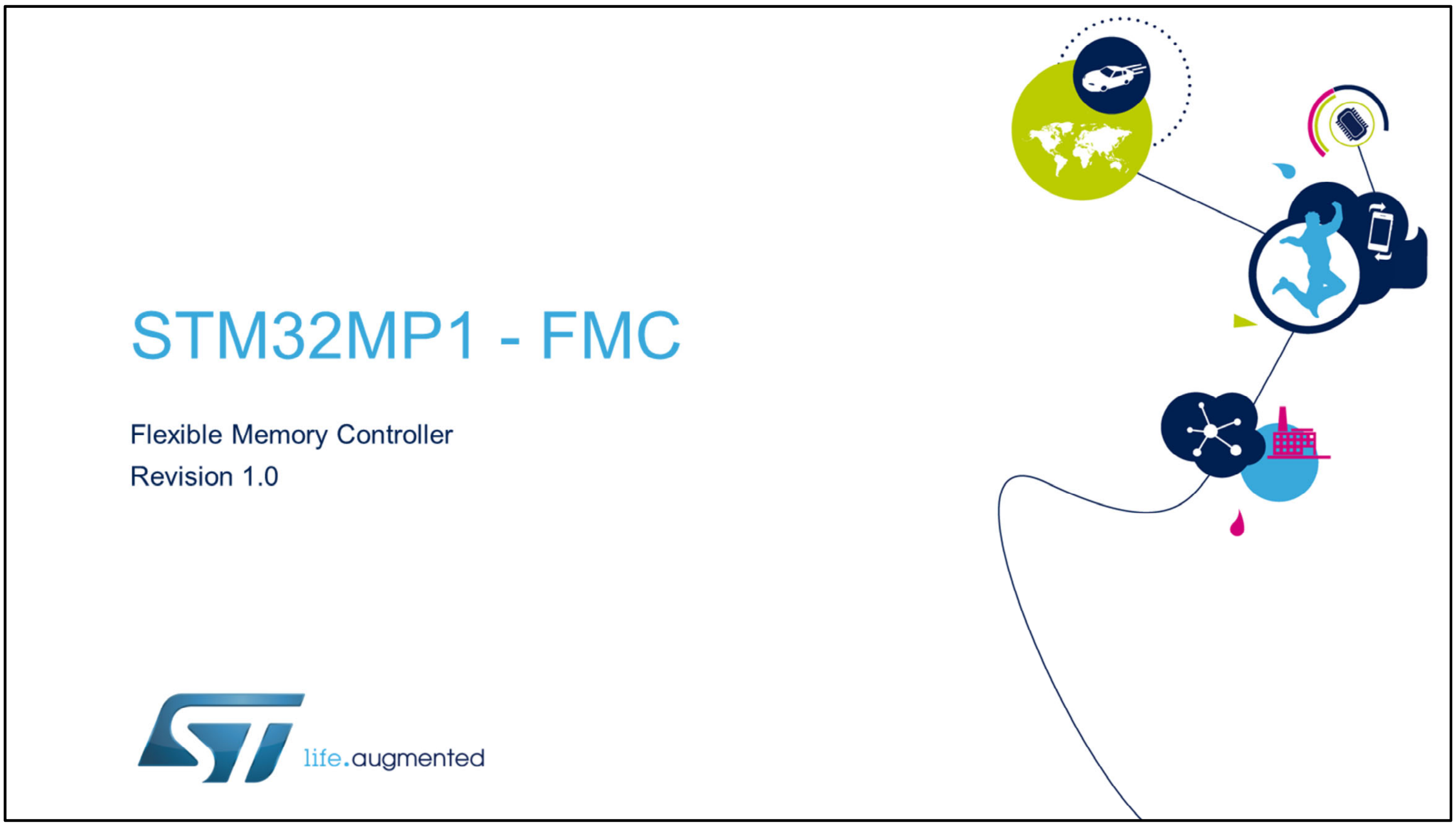


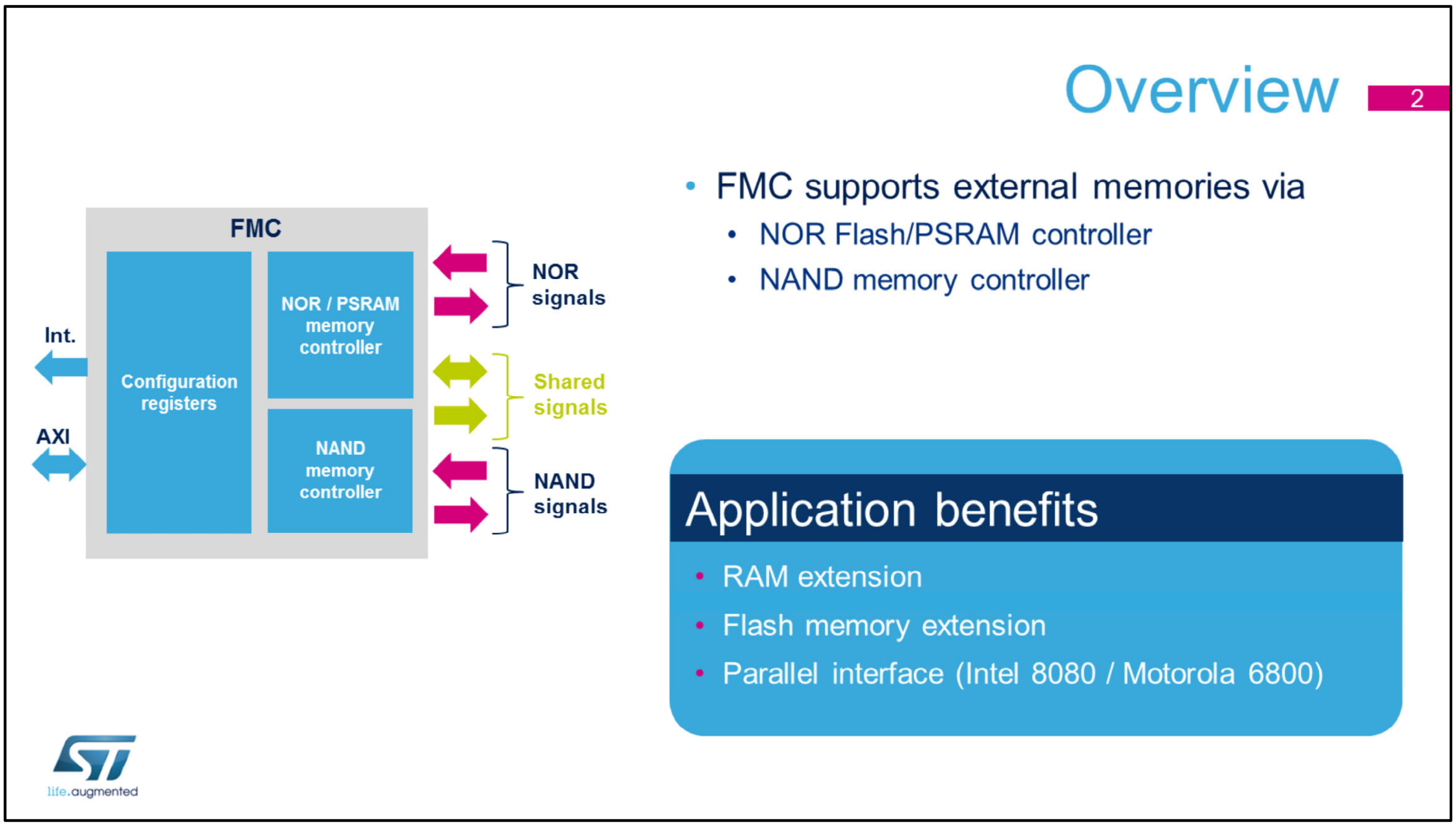
STM32MP1 - FMC

Flexible Memory Controller

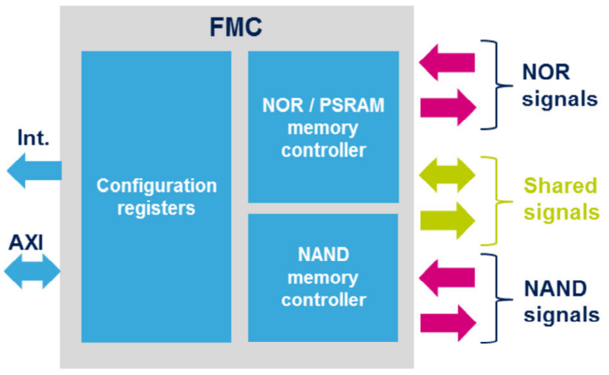
Revision 1.0

life.augmented

Hello, and welcome to this presentation of the STM32 Flexible Memory Controller. It covers all the features of this interface which is used to connect external memories such as NOR Flash, SRAM, PSRAM and NAND Flash memories.



Overview

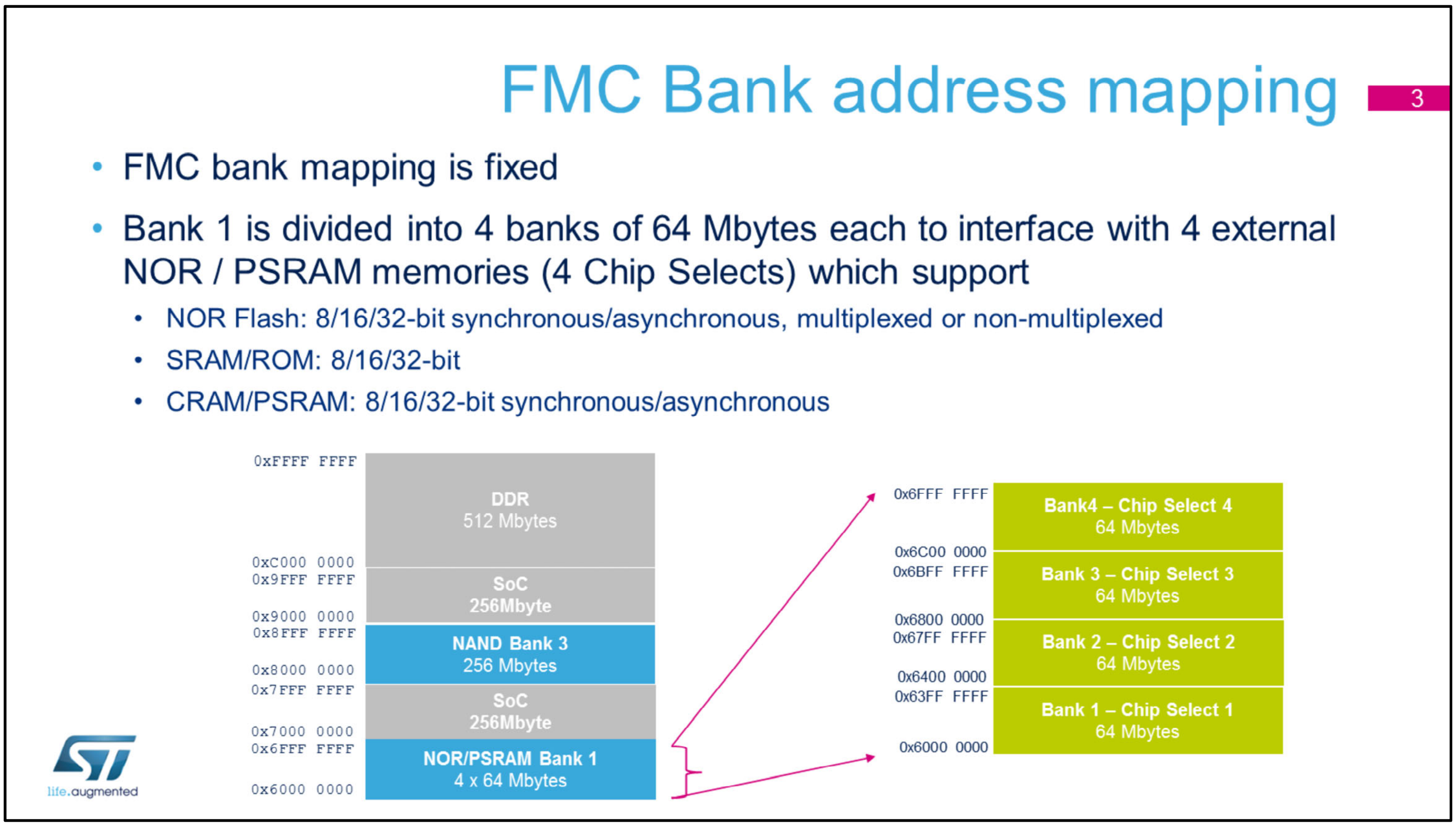


- FMC supports external memories via
  - NOR Flash/PSRAM controller
  - NAND memory controller

Application benefits

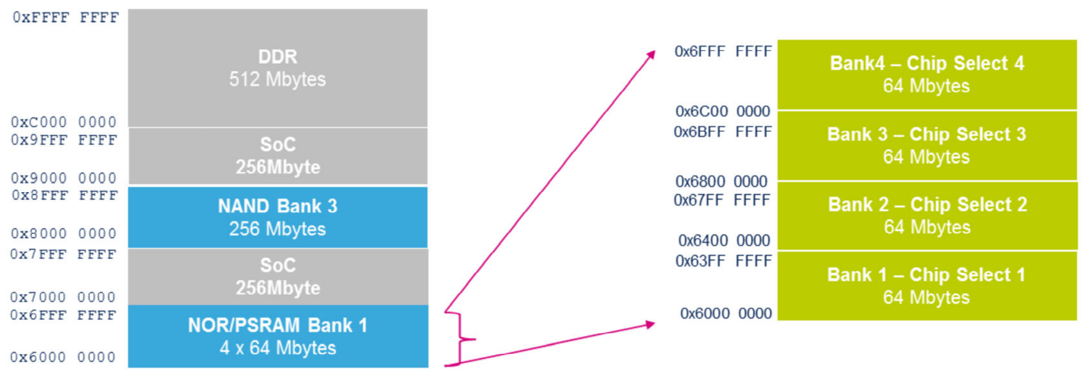
- RAM extension
- Flash memory extension
- Parallel interface (Intel 8080 / Motorola 6800)

The FMC controller integrated in STM32MP1 microprocessors provides external memory support through two memory controllers: The NOR Flash/PSRAM memory controller and the NAND Flash memory controller. This enables the CPU to communicate with external memories including NOR and NAND Flash memories, PSRAM and SRAM.
This interface is fully configurable, allowing easy connection with external memories or other parallel interfaces.
The benefits of the FMC controller include not only RAM and Flash memory space extension, but also the ability to interface seamlessly with most LCD controllers which support Intel 8080 and Motorola 6800 modes. This LCD parallel interface capability makes it easy to build cost-effective graphic applications using LCD modules containing embedded controllers or high-performance solutions using external controllers with dedicated acceleration.



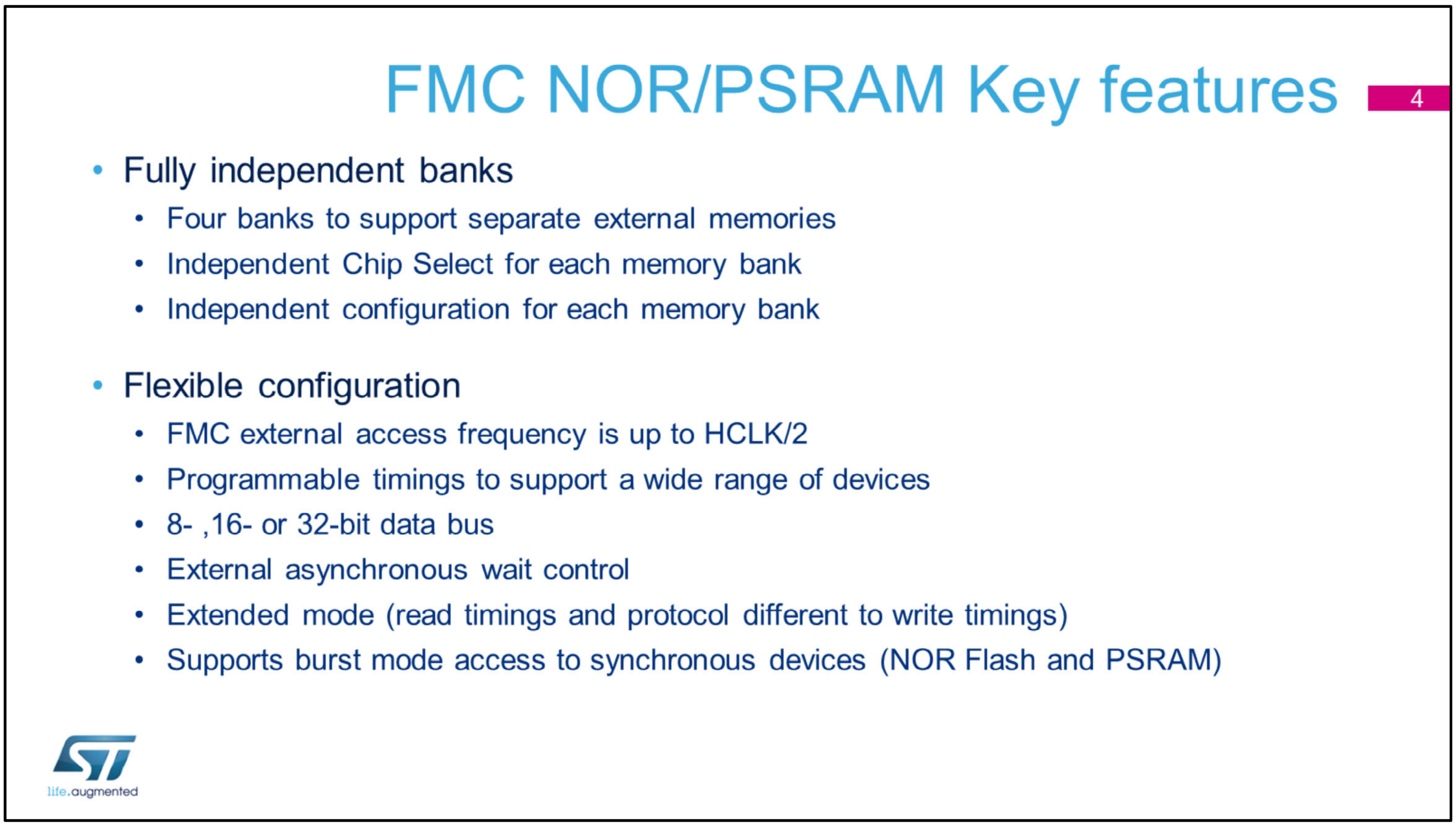
FMC Bank address mapping

- FMC bank mapping is fixed
- Bank 1 is divided into 4 banks of 64 Mbytes each to interface with 4 external NOR / PSRAM memories (4 Chip Selects) which support
  - NOR Flash: 8/16/32-bit synchronous/asynchronous, multiplexed or non-multiplexed
  - SRAM/ROM: 8/16/32-bit
  - CRAM/PSRAM: 8/16/32-bit synchronous/asynchronous



The mapping of the FMC bank addresses is fixed.
- Bank 1 is used by the NOR/PSRAM memory controller.
- Bank 3 is used by the NAND memory controller.

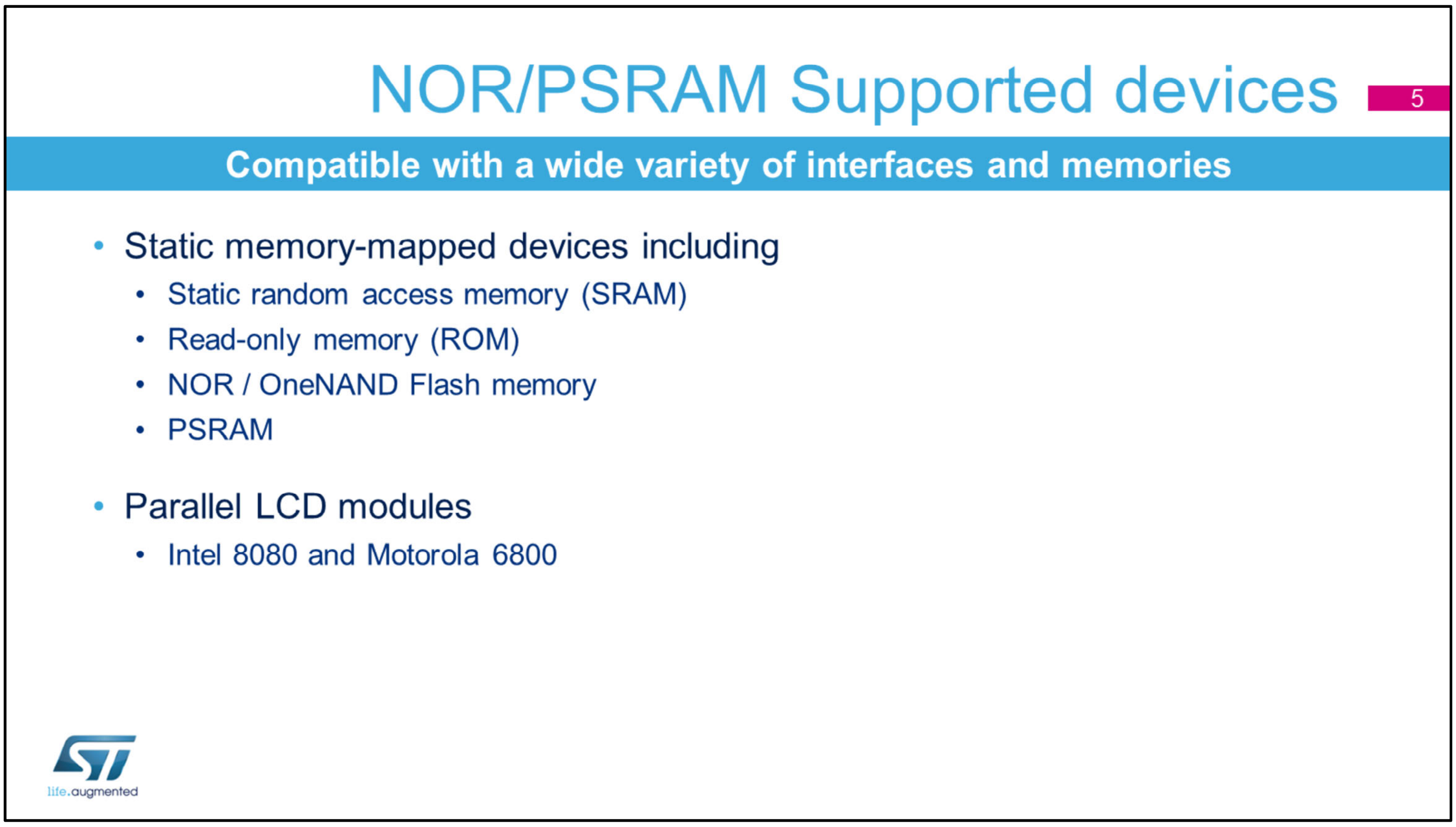
All other Banks are not used by the flexible memory controller and are available to the SoC memory map.



FMC NOR/PSRAM Key features

- Fully independent banks
    - Four banks to support separate external memories
    - Independent Chip Select for each memory bank
    - Independent configuration for each memory bank
- Flexible configuration
    - FMC external access frequency is up to HCLK/2
    - Programmable timings to support a wide range of devices
    - 8- ,16- or 32-bit data bus
    - External asynchronous wait control
    - Extended mode (read timings and protocol different to write timings)
    - Supports burst mode access to synchronous devices (NOR Flash and PSRAM)

The FMC controller offers four independent banks to support separate external memories. Each bank has an independent Chip Select and an independent configuration.
Each bank features programmable timings, a configurable 8- 16- or 32-bit data bus, and can access memory in asynchronous or burst mode for synchronous memory such as NOR Flash and PSRAM.
Synchronous memory can be accessed at a maximum frequency of HCLK divided by 2.

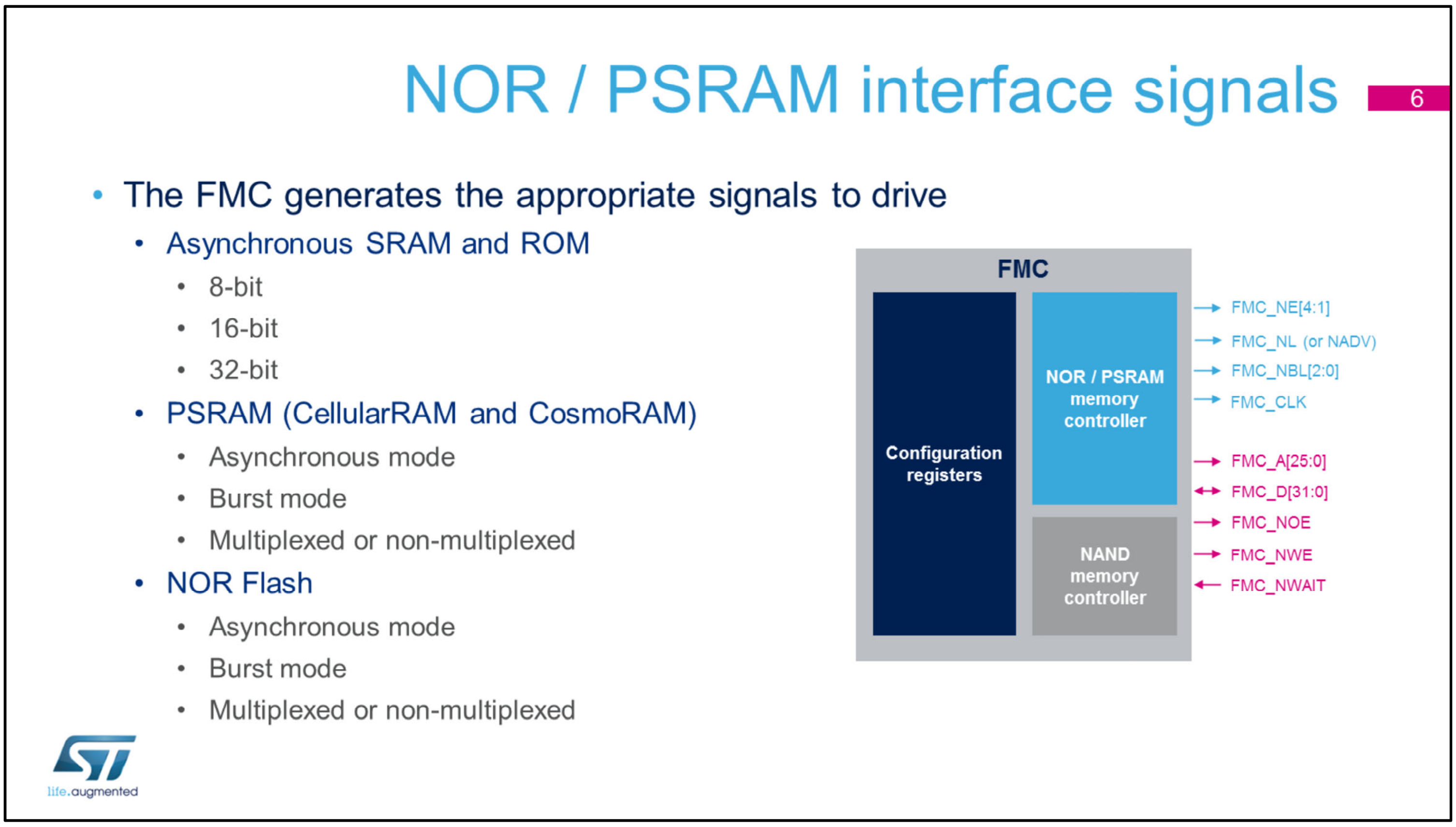


NOR/PSRAM Supported devices

Compatible with a wide variety of interfaces and memories

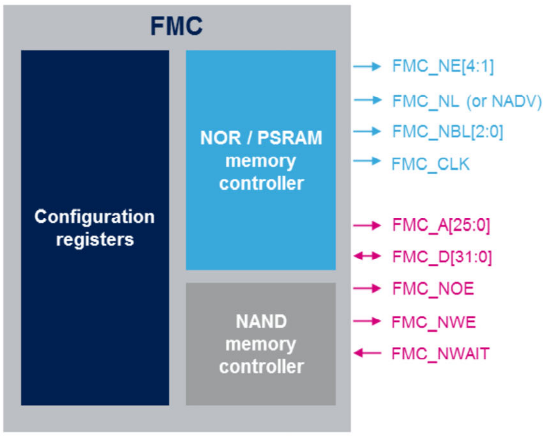
- Static memory-mapped devices including
  - Static random access memory (SRAM)
  - Read-only memory (ROM)
  - NOR / OneNAND Flash memory
  - PSRAM
- Parallel LCD modules
  - Intel 8080 and Motorola 6800

The FMC controller supports a wide variety of devices and memories.
It interfaces with static memory-mapped including static random access memory (SRAM), read-only memory (ROM), NOR / OneNAND Flash memory and PSRAM.
Furthermore, the FMC interface with parallel LCD modules, supporting the Intel 8080 and Motorola 6800 modes, and is flexible enough to adapt to various LCD interfaces.



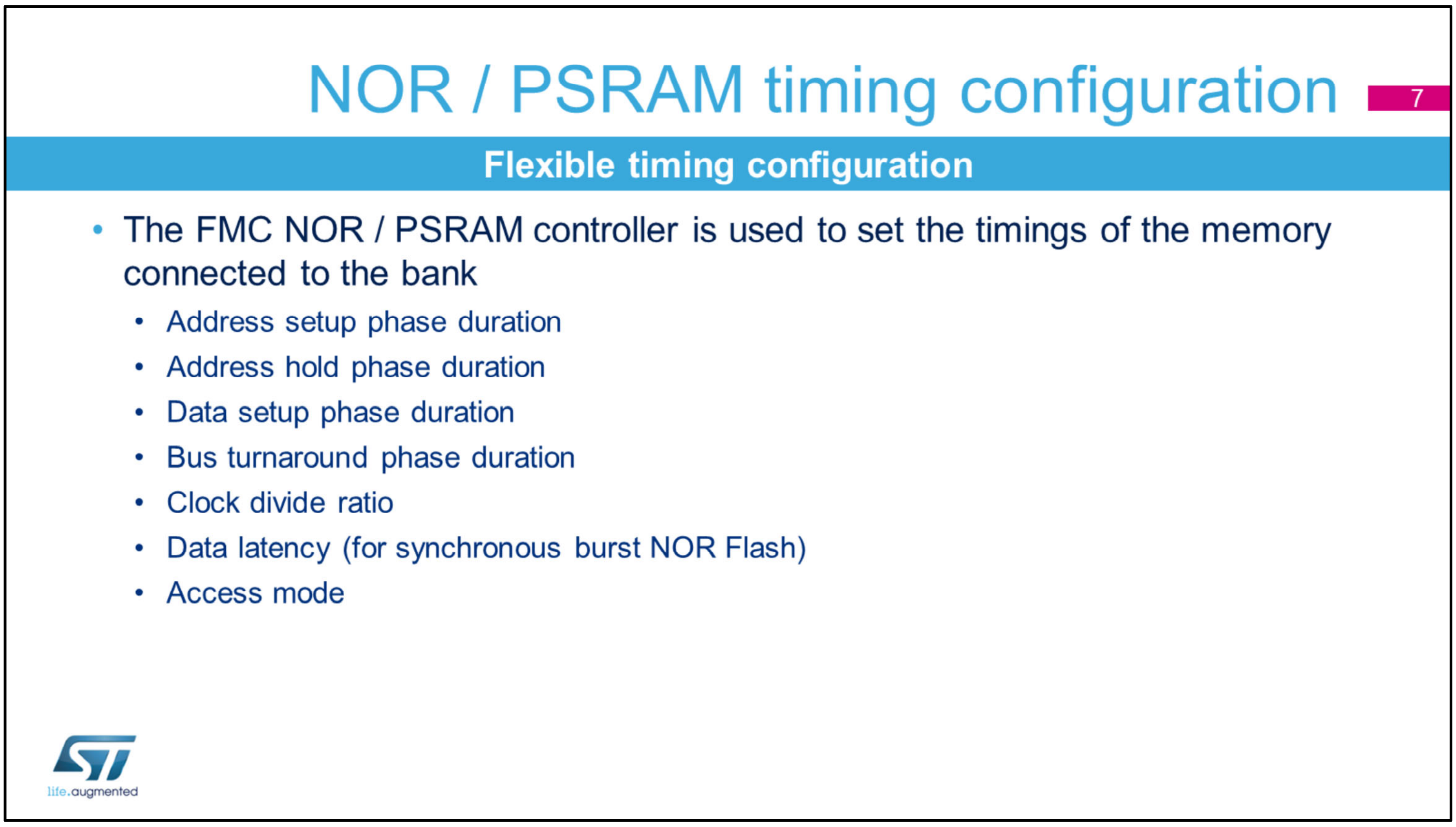
NOR / PSRAM interface signals

- The FMC generates the appropriate signals to drive
  - Asynchronous SRAM and ROM
    - 8-bit
    - 16-bit
    - 32-bit
  - PSRAM (CellularRAM and CosmoRAM)
    - Asynchronous mode
    - Burst mode
    - Multiplexed or non-multiplexed
  - NOR Flash
    - Asynchronous mode
    - Burst mode
    - Multiplexed or non-multiplexed



life.augmented

The FMC outputs a unique Chip Select signal to each bank and performs only one access at a time to an external device. The external memories are connected either to the NOR PSRAM controller or the NAND controller, and share address, data, and control signals.



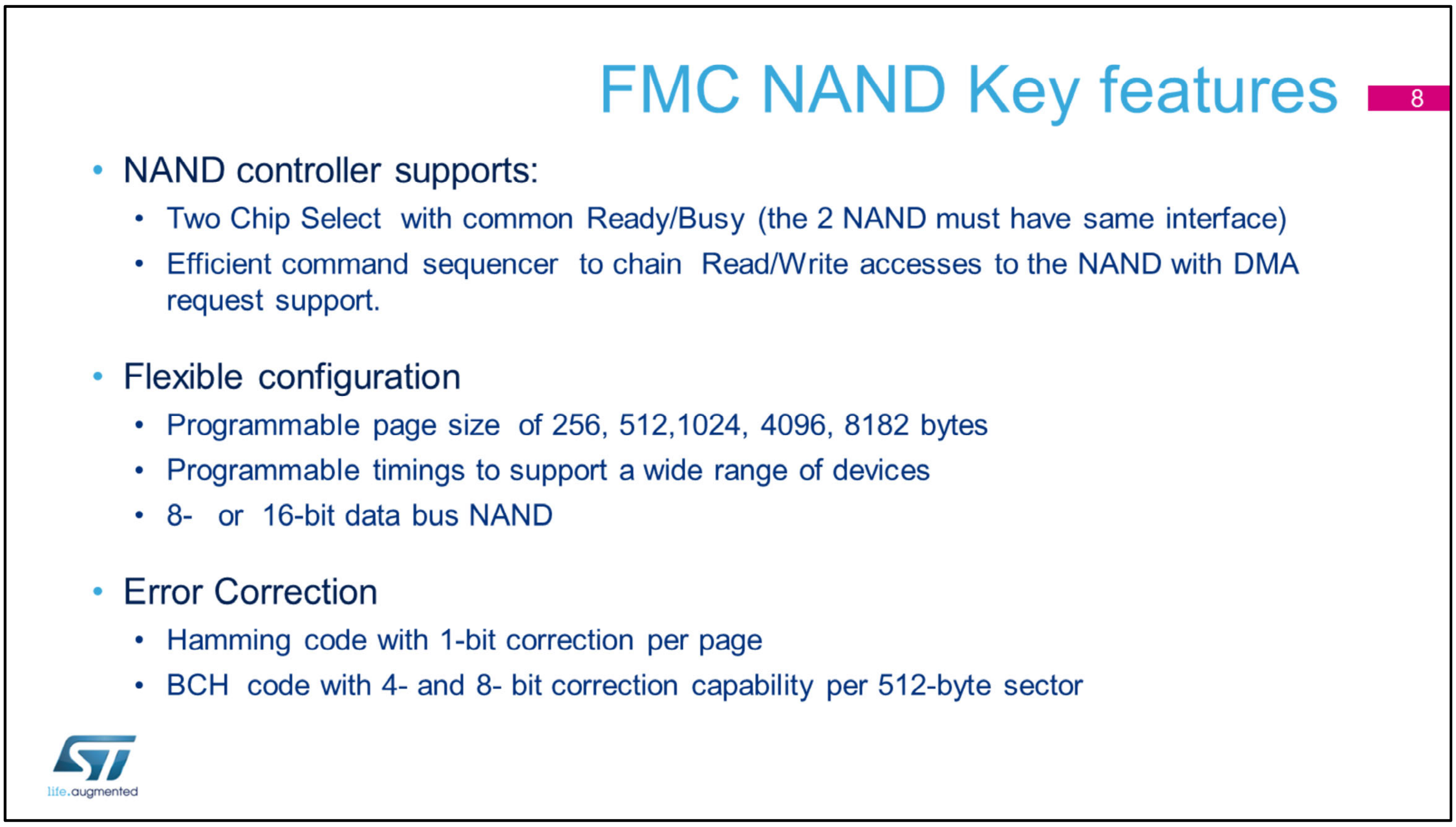
NOR / PSRAM timing configuration

Flexible timing configuration

- The FMC NOR / PSRAM controller is used to set the timings of the memory connected to the bank
  - Address setup phase duration
  - Address hold phase duration
  - Data setup phase duration
  - Bus turnaround phase duration
  - Clock divide ratio
  - Data latency (for synchronous burst NOR Flash)
  - Access mode

The NOR PSRAM controller allows the configuration of various timing parameters for the supported memories:

- Address setup phase: Duration of the first access phase
- Address hold phase: Duration of the middle phase of the access cycle
- Data setup phase: Duration of the second access phase
- Bus turnaround phase: Duration of the bus turnaround phase
- Clock divide ratio: Number of AHB clock cycles (HCLK) within one memory clock cycle (CLK)
- Data latency: Number of clock cycles to be issued to the memory before the first data transfer
- Access mode.



FMC NAND Key features

8

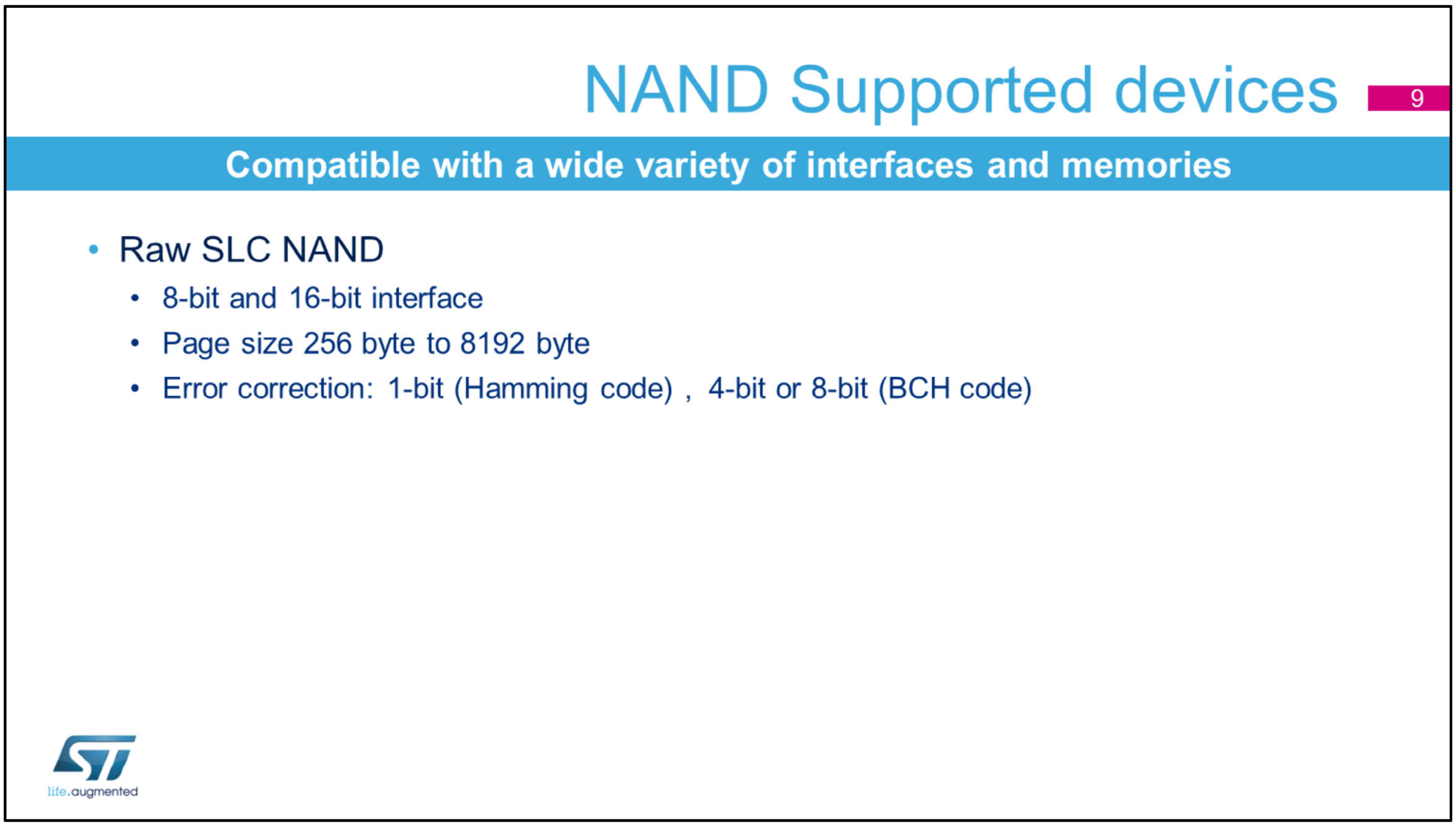
- NAND controller supports:
  - Two Chip Select with common Ready/Busy (the 2 NAND must have same interface)
  - Efficient command sequencer to chain Read/Write accesses to the NAND with DMA request support.
- Flexible configuration
  - Programmable page size of 256, 512,1024, 4096, 8182 bytes
  - Programmable timings to support a wide range of devices
  - 8- or 16-bit data bus NAND
- Error Correction
  - Hamming code with 1-bit correction per page
  - BCH code with 4- and 8- bit correction capability per 512-byte sector

The FMC controller features a NAND memory controller supporting

- Up to 2 NAND devices (of the same type) and a common Read/Busy signal
- Programmable page size up to 8Kbyte
- Programmable timing parameters
- 8-bit or 16-bit interface

The NAND memory controller has an hardware error detection and correction supporting

- Hamming code (1-bit correction per page)
- BCH code with either 4-bit correction or 8-bit correction per 512-byte sector.



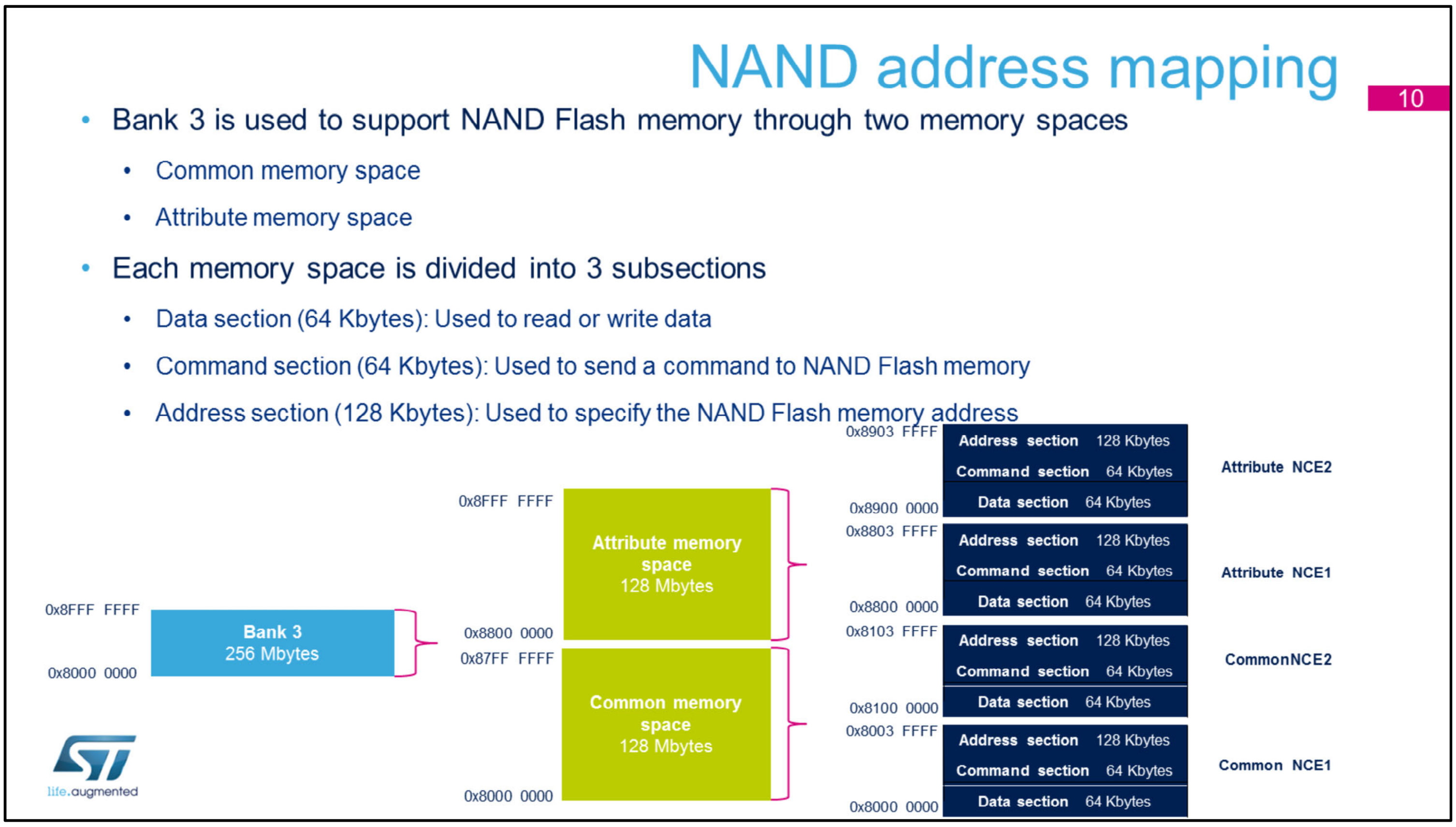
NAND Supported devices

Compatible with a wide variety of interfaces and memories

- Raw SLC NAND
  - 8-bit and 16-bit interface
  - Page size 256 byte to 8192 byte
  - Error correction: 1-bit (Hamming code) , 4-bit or 8-bit (BCH code)

life.augmented

The FMC also interfaces with NAND Flash memories and supports error correction code (ECC) for up to 8 Kbytes of data read or written. Three interrupt sources can be configured to generate an interrupt when a rising edge, falling edge, or high level is detected on the NAND Flash Ready/Busy signal.



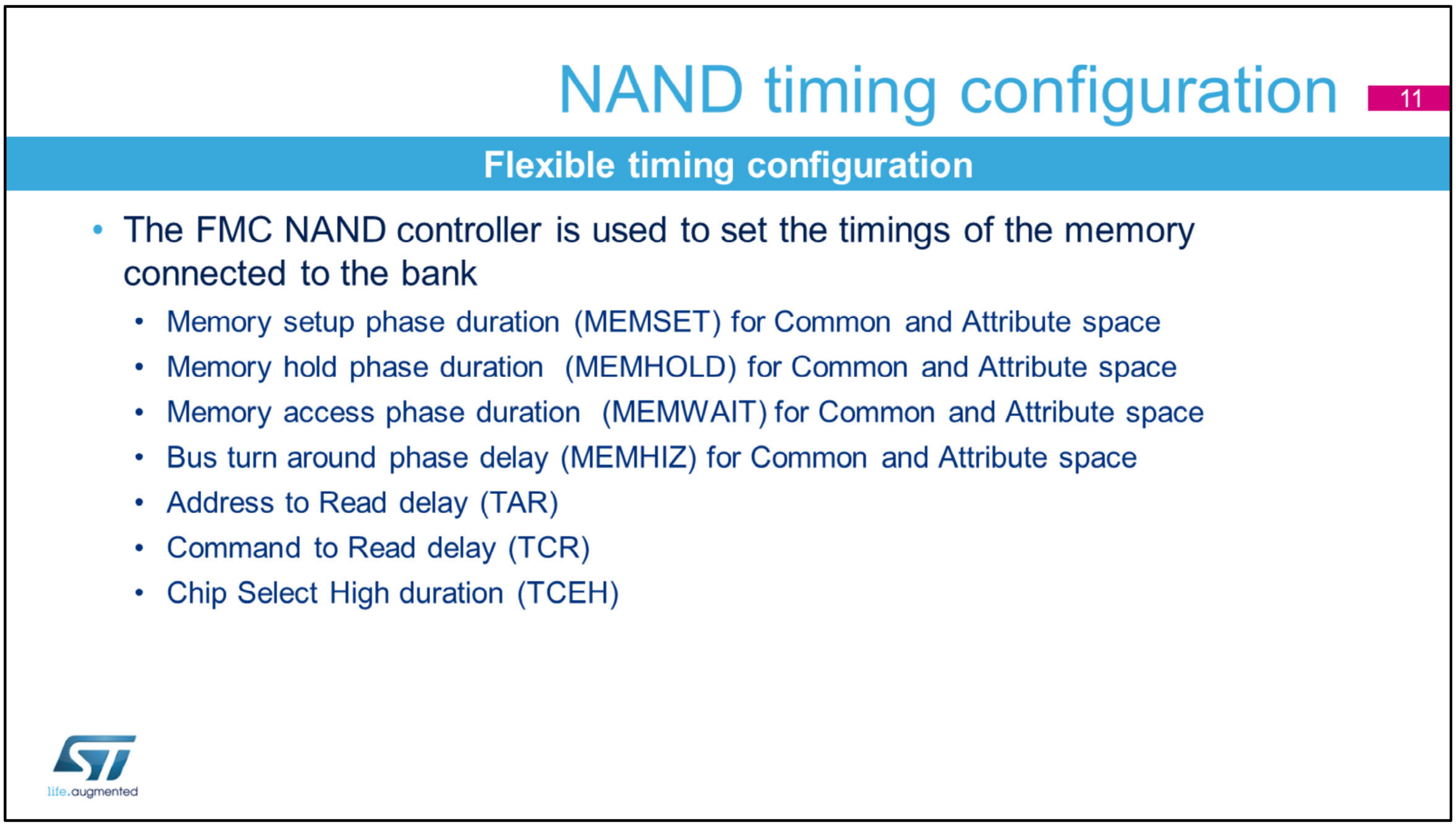
NAND address mapping

- Bank 3 is used to support NAND Flash memory through two memory spaces
  - Common memory space
  - Attribute memory space
- Each memory space is divided into 3 subsections
  - Data section (64 Kbytes): Used to read or write data
  - Command section (64 Kbytes): Used to send a command to NAND Flash memory
  - Address section (128 Kbytes): Used to specify the NAND Flash memory address

Bank 3 is used to interface with the NAND Flash memory. It is divided into two memory spaces: Common memory space and Attribute memory space. Both spaces are similar. The common memory space is for all NAND Flash read and write accesses, except when writing the last address byte to the NAND Flash device, where the CPU must write to the attribute memory space. This allows to implement the pre-wait functionality needed by certain NAND Flash memories by writing the last address byte with different timings.
Each memory space is subdivided into three sections:

- Data section (64 Kbytes): Used to read or write data from NAND Flash memory.
- Command section (64 Kbytes): Used to send a command to NAND Flash memory.
- Address section (128 Kbytes): Used to specify the NAND Flash memory address.

The NAND device (NCE1 and NCE2) is decoded according to the address bit 24 (16MByte range).



NAND timing configuration

Flexible timing configuration

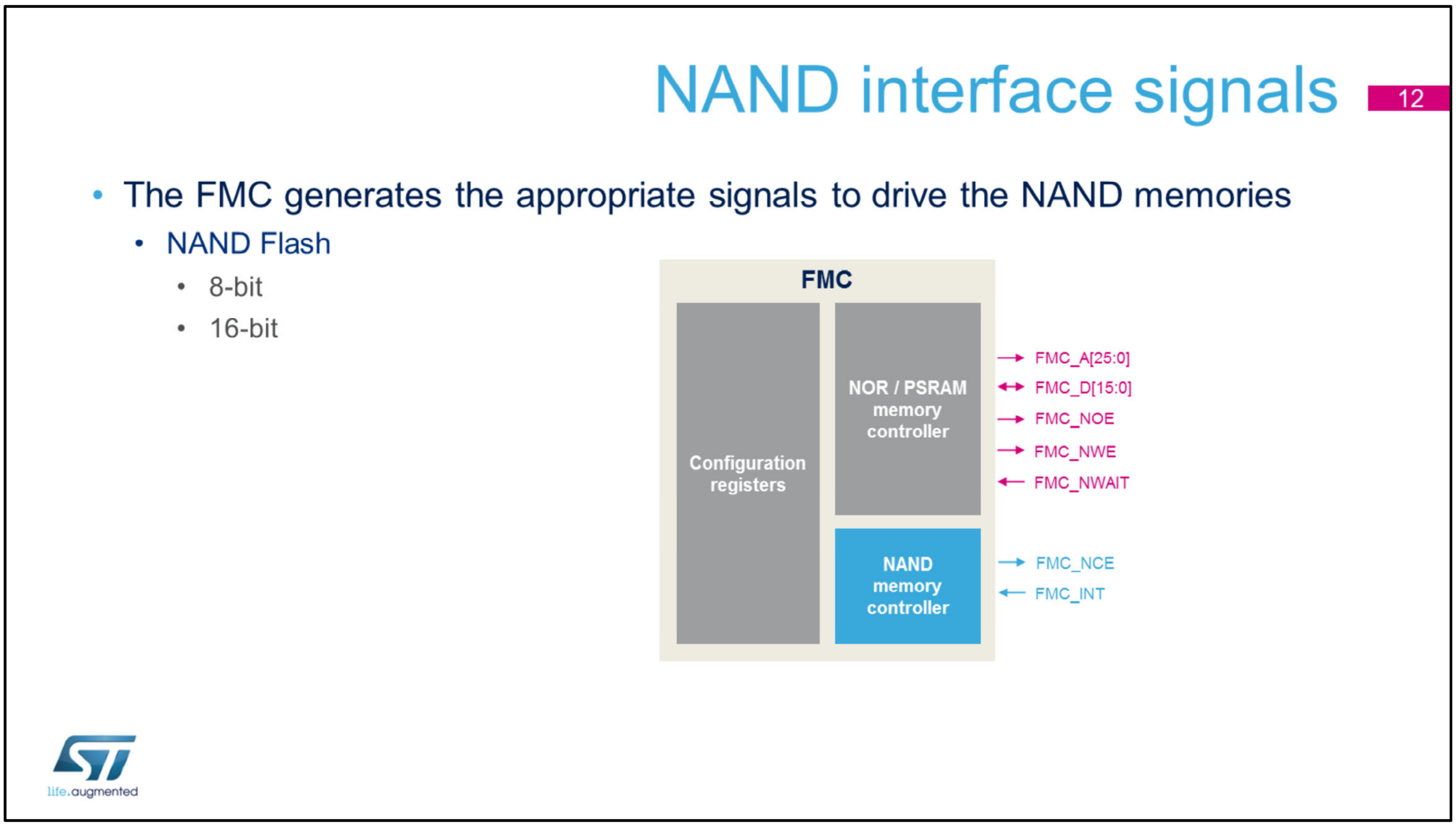
- The FMC NAND controller is used to set the timings of the memory connected to the bank
  - Memory setup phase duration (MEMSET) for Common and Attribute space
  - Memory hold phase duration (MEMHOLD) for Common and Attribute space
  - Memory access phase duration (MEMWAIT) for Common and Attribute space
  - Bus turn around phase delay (MEMHIZ) for Common and Attribute space
  - Address to Read delay (TAR)
  - Command to Read delay (TCR)
  - Chip Select High duration (TCEH)

Each common and attribute memory space can be configured with different timings for the NAND Flash’s command, address write, and data read/write accesses.
The attribute memory space is used for the last address write access if the timing must differ from that of previous accesses in case of Ready/Busy management. Otherwise, only common space is needed.
Four parameters are used to define the number of HCLK cycles for the different phases of any NAND Flash access:

- Setup time
- Hold time
- Wait time
- Data bus HiZ time

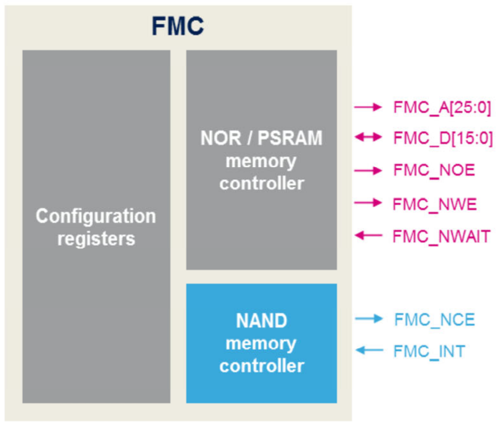
3 additional parameters are used to control the timings for:

- The Address to Read delay
- The Command to Read delay
- The minimum Chip Select High duration

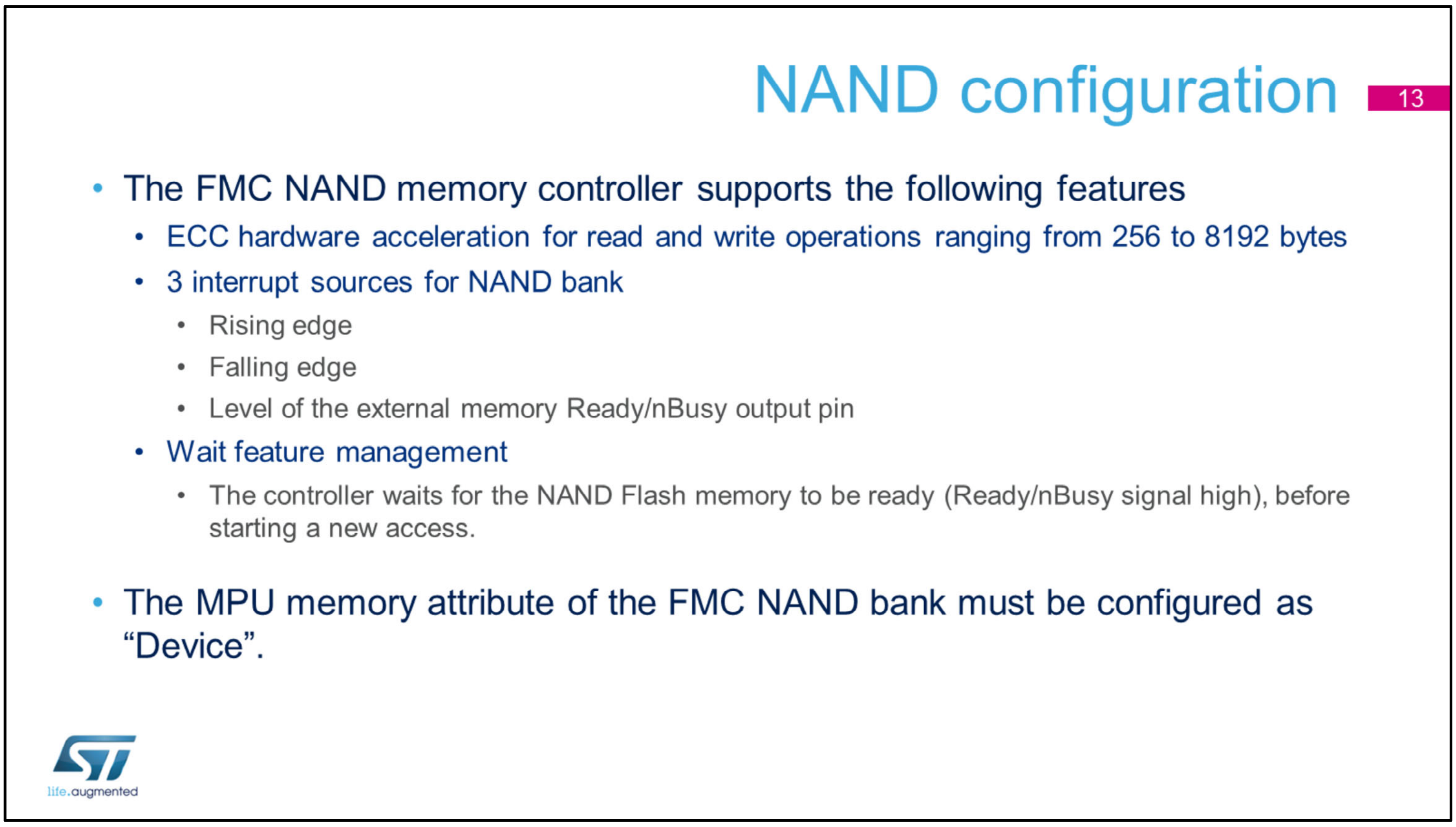


NAND interface signals

- The FMC generates the appropriate signals to drive the NAND memories
  - NAND Flash
    - 8-bit
    - 16-bit



The FMC generates the appropriate signals to drive NAND Flash memory. The address, data, and control signals are shared with the NOR / PSRAM controller.
The command latch enable (CLE) and address latch enable (ALE) signals of the NAND Flash memory device are driven by address signals from the FMC controller connected to Address line 16 and Address line 17 respectively.
The ALE is active when writing to the address section and the CLE is active when writing to the command section.



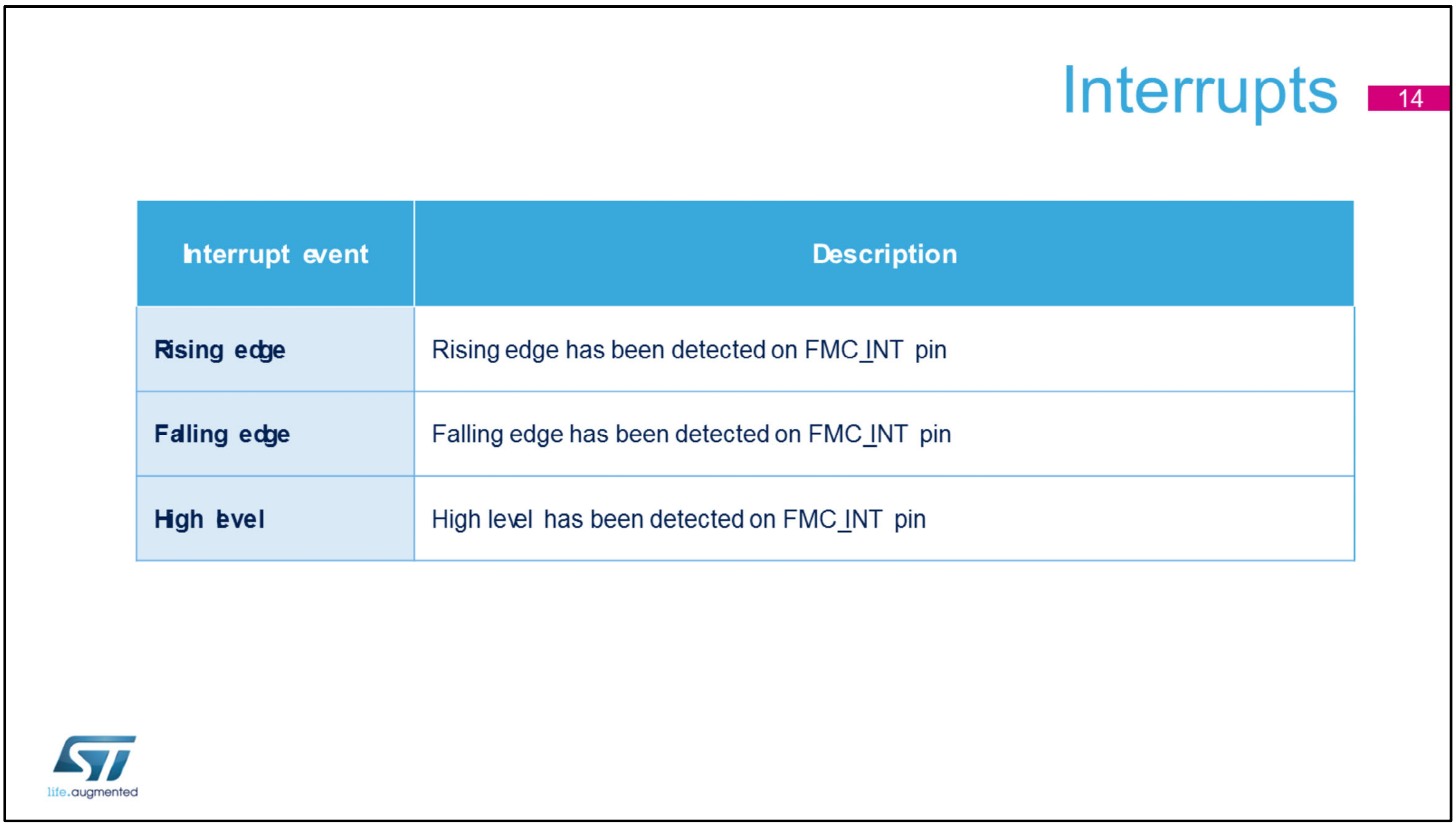
NAND configuration

13

- The FMC NAND memory controller supports the following features
  - ECC hardware acceleration for read and write operations ranging from 256 to 8192 bytes
  - 3 interrupt sources for NAND bank
    - Rising edge
    - Falling edge
    - Level of the external memory Ready/nBusy output pin
  - Wait feature management
    - The controller waits for the NAND Flash memory to be ready (Ready/nBusy signal high), before starting a new access.
- The MPU memory attribute of the FMC NAND bank must be configured as "Device".

life.augmented

The FMC NAND memory controller includes support for the following features:
Error correction code: The ECC algorithm can perform 1-bit error correction and 2-bit error detection per 256 to 8192 bytes read or written from/to the NAND Flash memory. It is based on the Hamming coding algorithm.
3 interrupt sources can be enabled to detect a rising edge, falling edge or level on Ready/Busy signal output from NAND Flash memory.
Wait feature management: The controller waits for the NAND Flash memory to be ready before starting a new access.
The MPU memory attribute of the FMC NAND bank must be configured as a Device

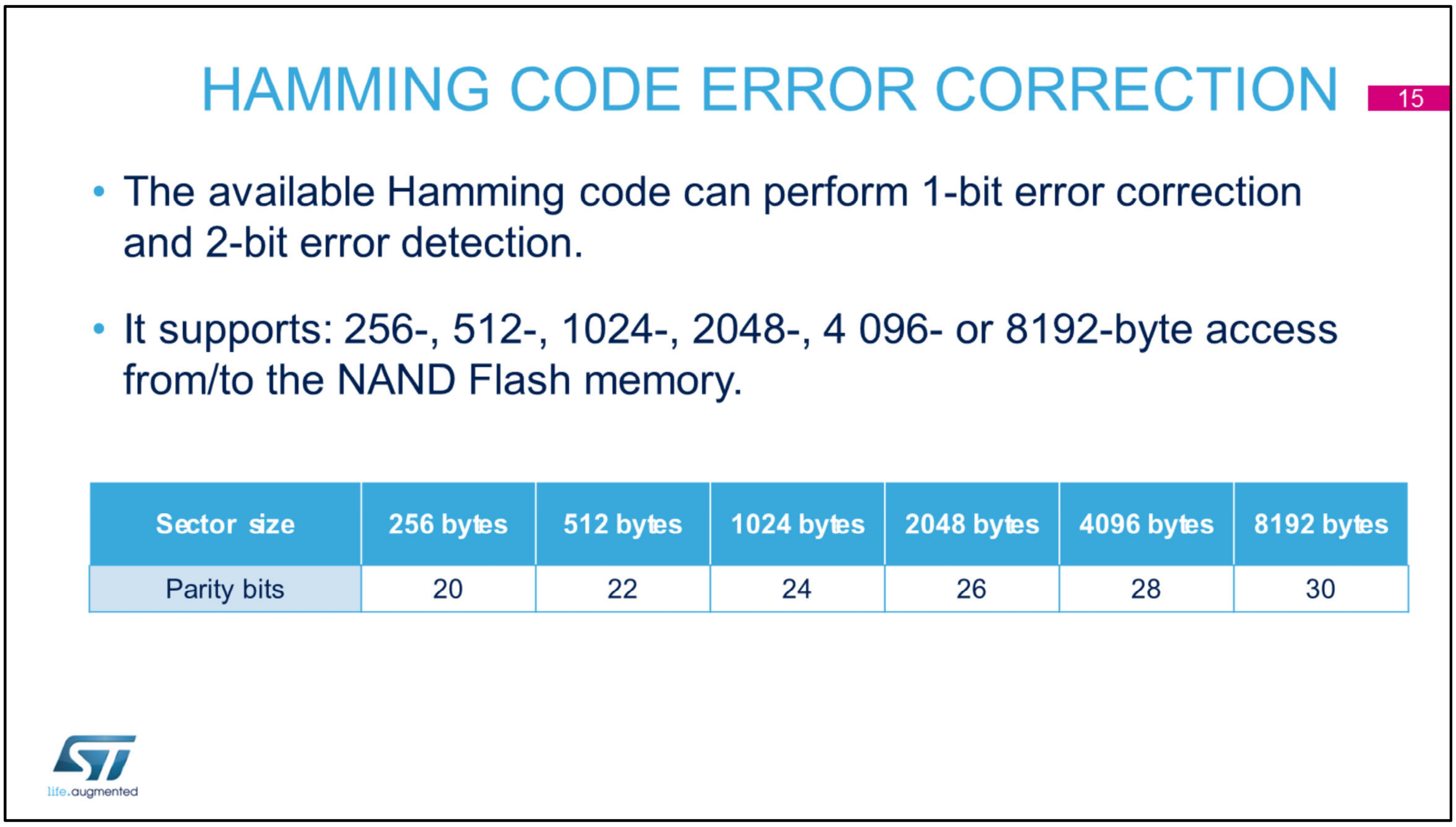


Interrupts

Interrupt event	Description
Rising edge	Rising edge has been detected on FMC_INT pin
Falling edge	Falling edge has been detected on FMC_INT pin
High level	High level has been detected on FMC_INT pin

life.augmented

The NAND controller offers 3 interrupt sources: rising edge, falling edge, and high level detection on the FSMC INT pin when it is connected to the Ready/nBusy signal from the NAND Flash memory.



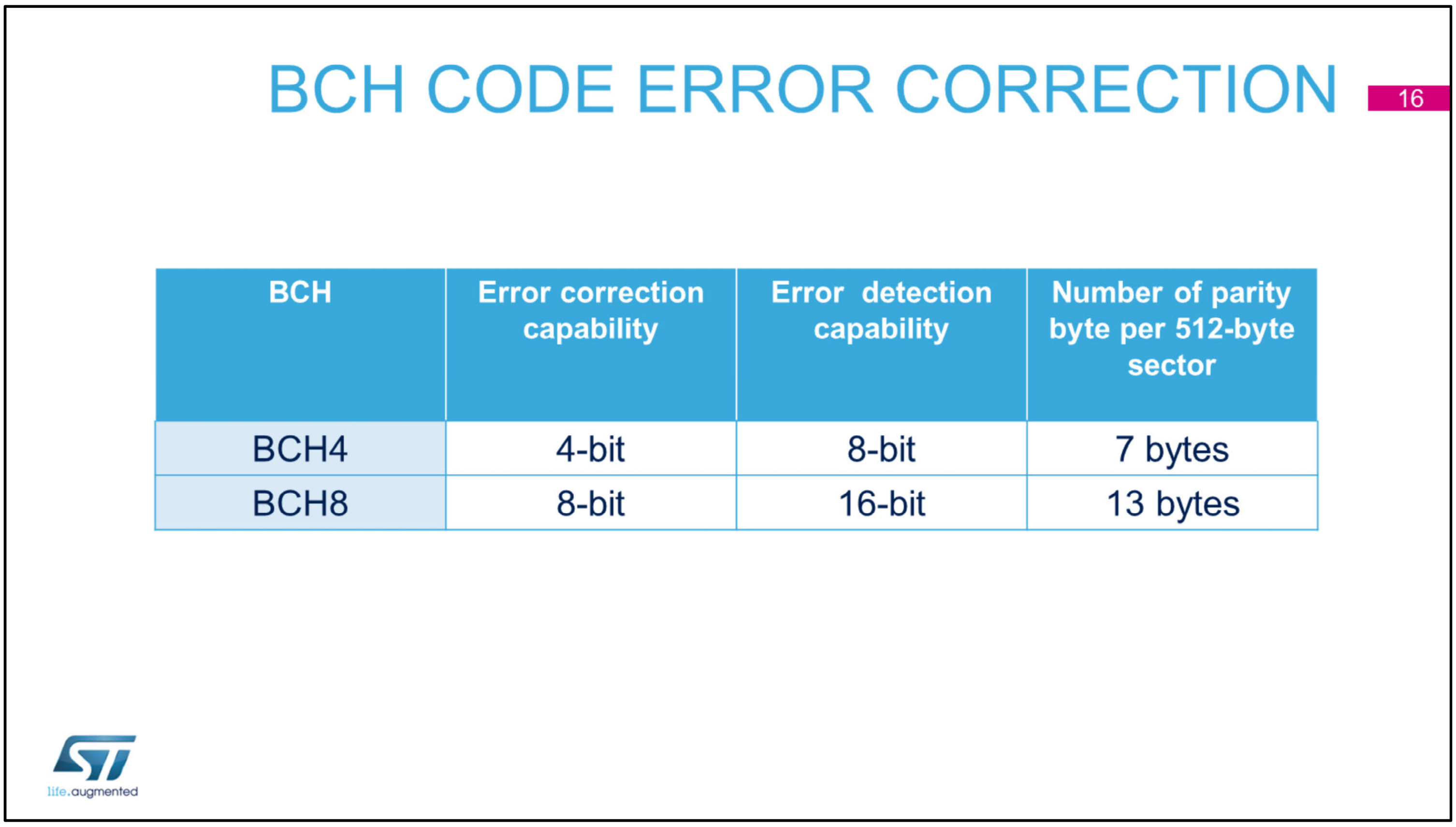
HAMMING CODE ERROR CORRECTION

- The available Hamming code can perform 1-bit error correction and 2-bit error detection.
- It supports: 256-, 512-, 1024-, 2048-, 4 096- or 8192-byte access from/to the NAND Flash memory.

Sector size	256 bytes	512 bytes	1024 bytes	2048 bytes	4096 bytes	8192 bytes
Parity bits	20	22	24	26	28	30

life.augmented

The available Hamming code can perform 1-bit error correction and 2-bit error detection per 256-, 512-, 1024-, 2048-, 4 096- or 8192-byte access from/to the NAND Flash memory. It consists in calculating the row and column parity. This algorithm is supported by 8-bit and 16-bit NAND Flash memories.



BCH CODE ERROR CORRECTION

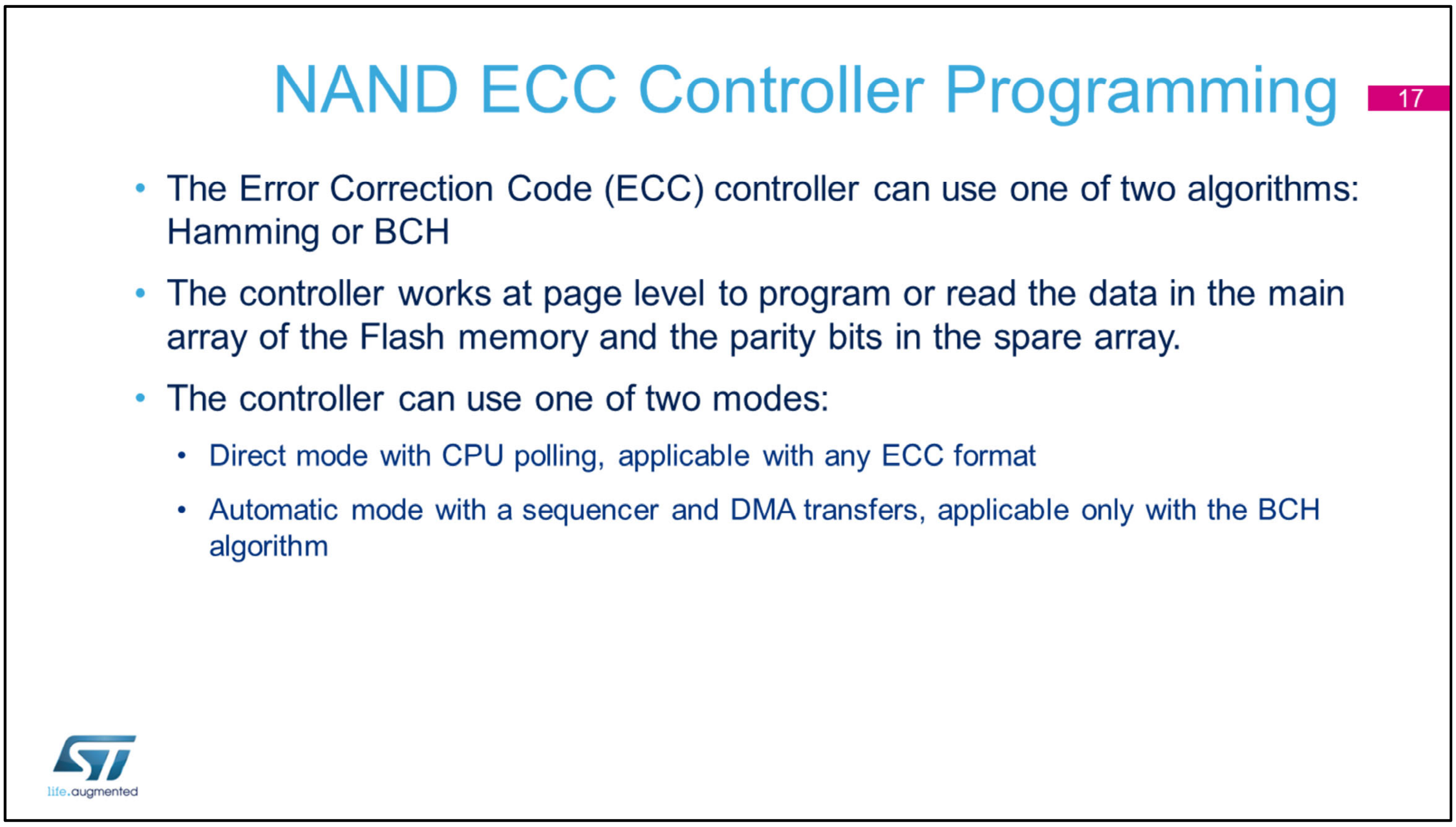
BCH	Error correction capability	Error detection capability	Number of parity byte per 512-byte sector
BCH4	4-bit	8-bit	7 bytes
BCH8	8-bit	16-bit	13 bytes

life.augmented

To increase the error correction capability, the FMC embeds a BCH (Bose, Chaudhuri and Hocquenghem) encoder and decoder.
It supports either:

- 4-bit error code correction with 8-bit error detection per sector
- 8-bit error code correction with 16-bit error detection per sector.

The BCH encoder/decoder module handles sectors of fixed size, equal to 512 bytes.



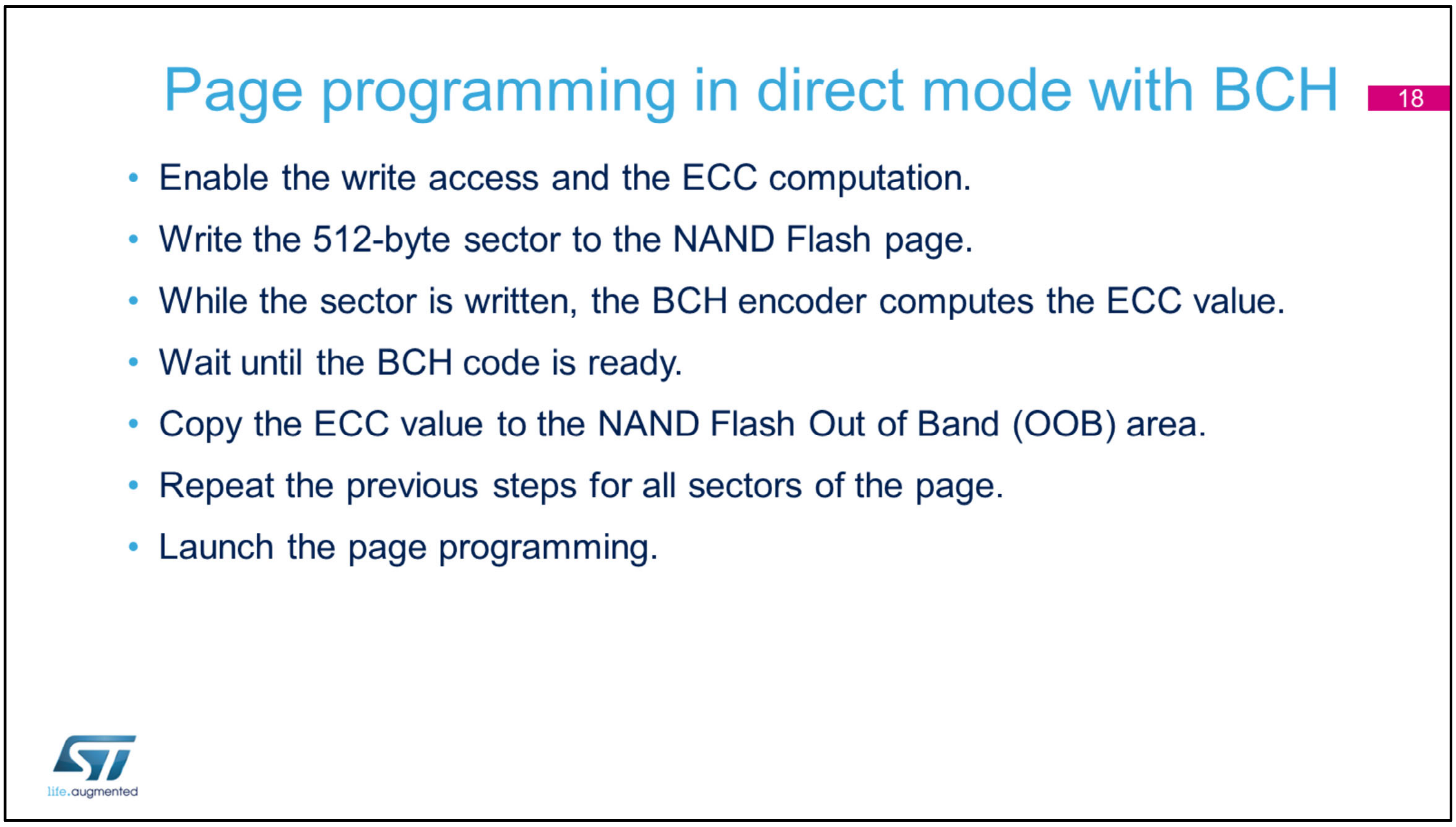
NAND ECC Controller Programming

- The Error Correction Code (ECC) controller can use one of two algorithms: Hamming or BCH
- The controller works at page level to program or read the data in the main array of the Flash memory and the parity bits in the spare array.
- The controller can use one of two modes:
  - Direct mode with CPU polling, applicable with any ECC format
  - Automatic mode with a sequencer and DMA transfers, applicable only with the BCH algorithm

life.augmented

The Error Correction Code (ECC) controller can use one of two algorithms: Hamming or BCH, depending on the ECC requirements of the NAND Flash device.
The controller works at page level for reading and/or programming the NAND Flash memory with data located in the main array and parity bits in the spare array (also known as OOB).
The controller can use one of two modes:

- Direct mode with CPU polling, applicable with any ECC,
- Automatic mode with a sequencer and DMA transfers, applicable only with the BCH algorithm.

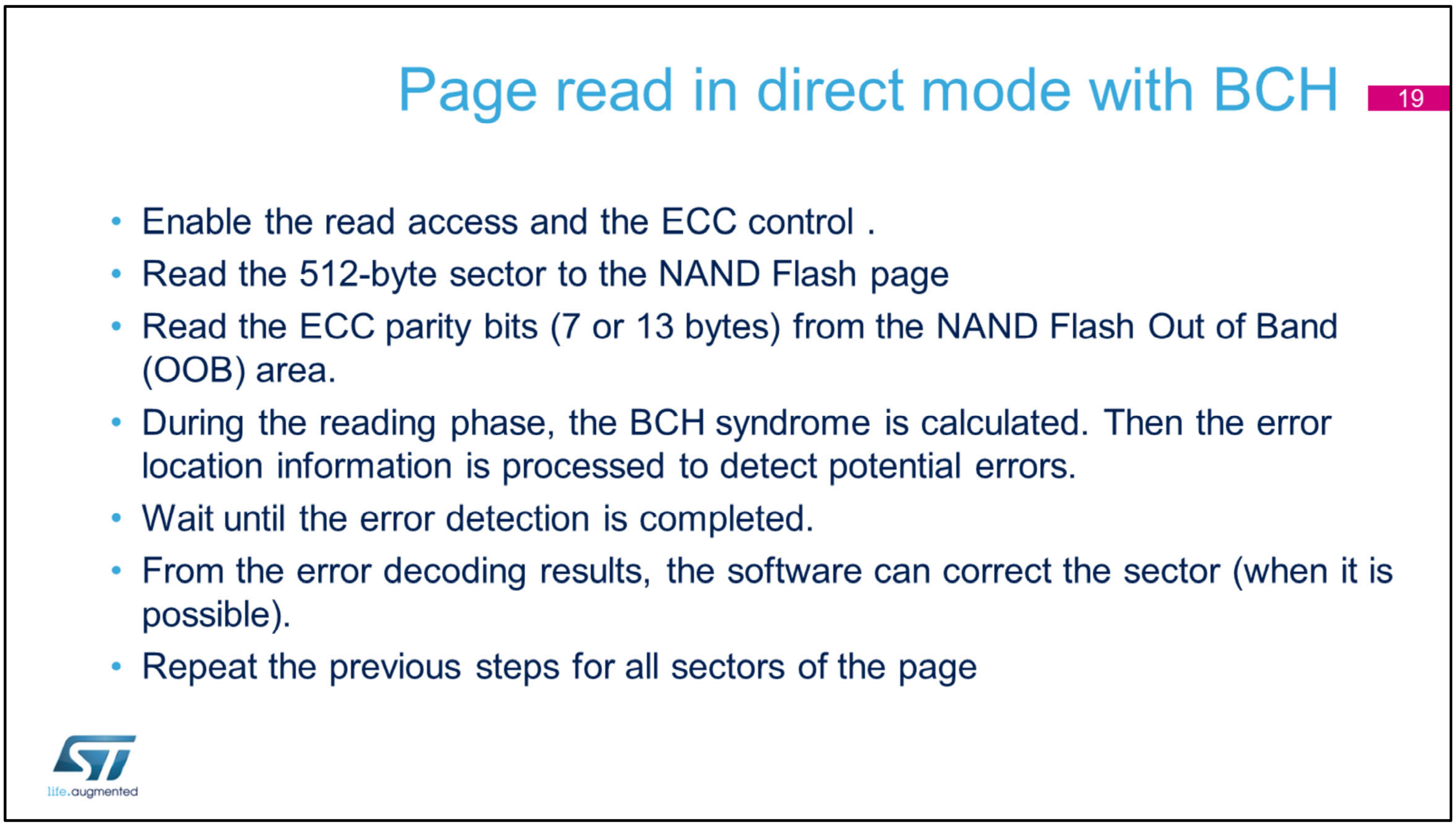


Page programming in direct mode with BCH

- Enable the write access and the ECC computation.
- Write the 512-byte sector to the NAND Flash page.
- While the sector is written, the BCH encoder computes the ECC value.
- Wait until the BCH code is ready.
- Copy the ECC value to the NAND Flash Out of Band (OOB) area.
- Repeat the previous steps for all sectors of the page.
- Launch the page programming.

To program a page in direct mode:

- Enable the write access and the ECC computation.
- Write the 512-byte sector to the NAND Flash page.
- While the sector is written, the BCH encoder computes the ECC value.
- Wait until the BCH code is ready.
- Copy the ECC value to the NAND Flash Out of Band (OOB) area.
- Repeat the previous steps for all sectors of the page.
- Launch the page programming.



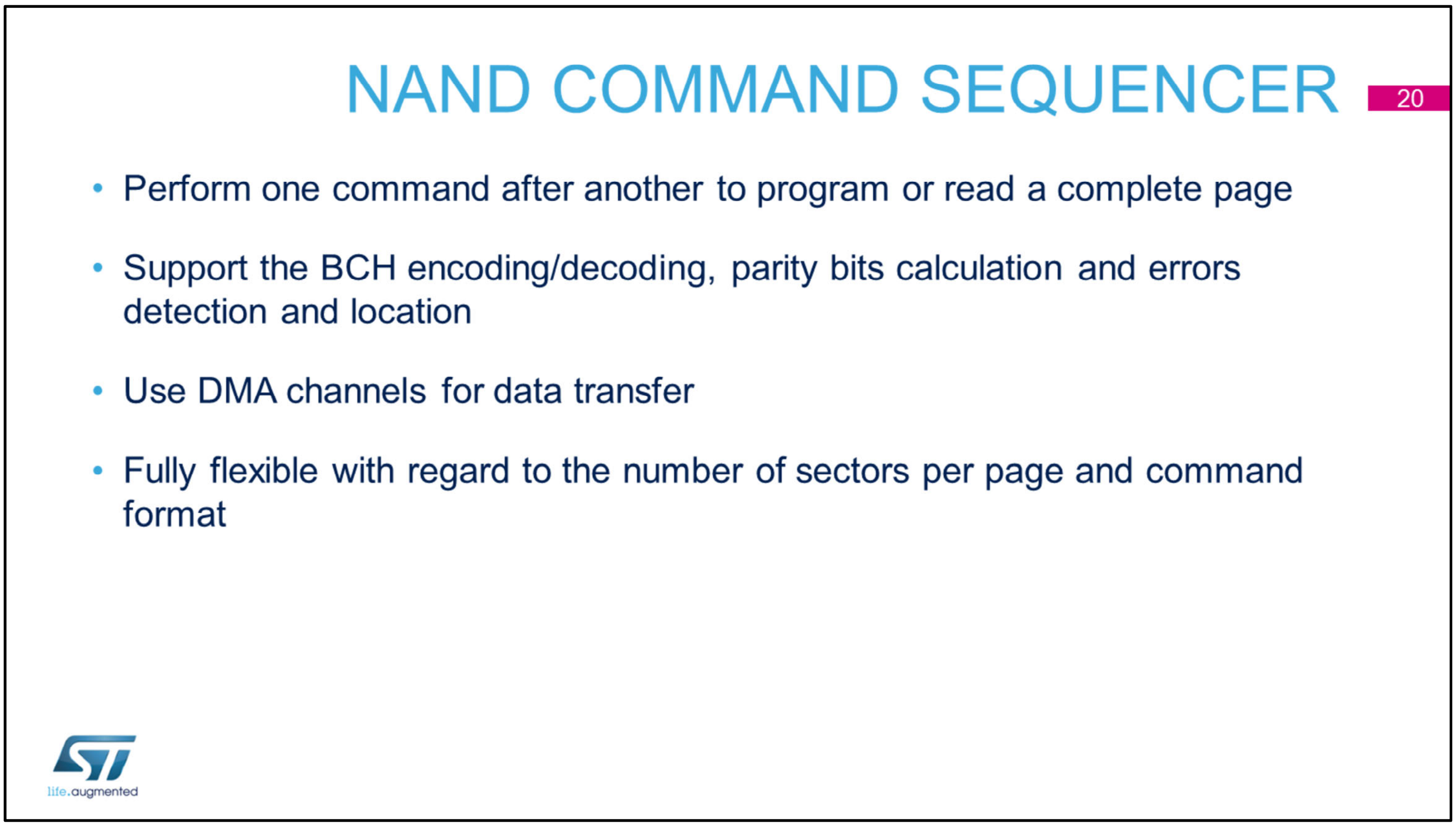
Page read in direct mode with BCH

- Enable the read access and the ECC control .
- Read the 512-byte sector to the NAND Flash page
- Read the ECC parity bits (7 or 13 bytes) from the NAND Flash Out of Band (OOB) area.
- During the reading phase, the BCH syndrome is calculated. Then the error location information is processed to detect potential errors.
- Wait until the error detection is completed.
- From the error decoding results, the software can correct the sector (when it is possible).
- Repeat the previous steps for all sectors of the page

life.augmented

To read a page in direct mode:

- Enable the read access and the ECC control.
- Read the 512-byte sector to the NAND Flash page.
- Read the ECC parity bits (7 or 13 bytes) from the NAND Flash Out of Band (OOB) area.
- During the reading phase, the BCH syndrome is calculated. Then the error location information is processed to detect potential errors.
- Wait until the error detection is completed.
- From the error decoding results, the software can correct the sector (when it is possible).
- Repeat the previous steps for all sectors of the page.

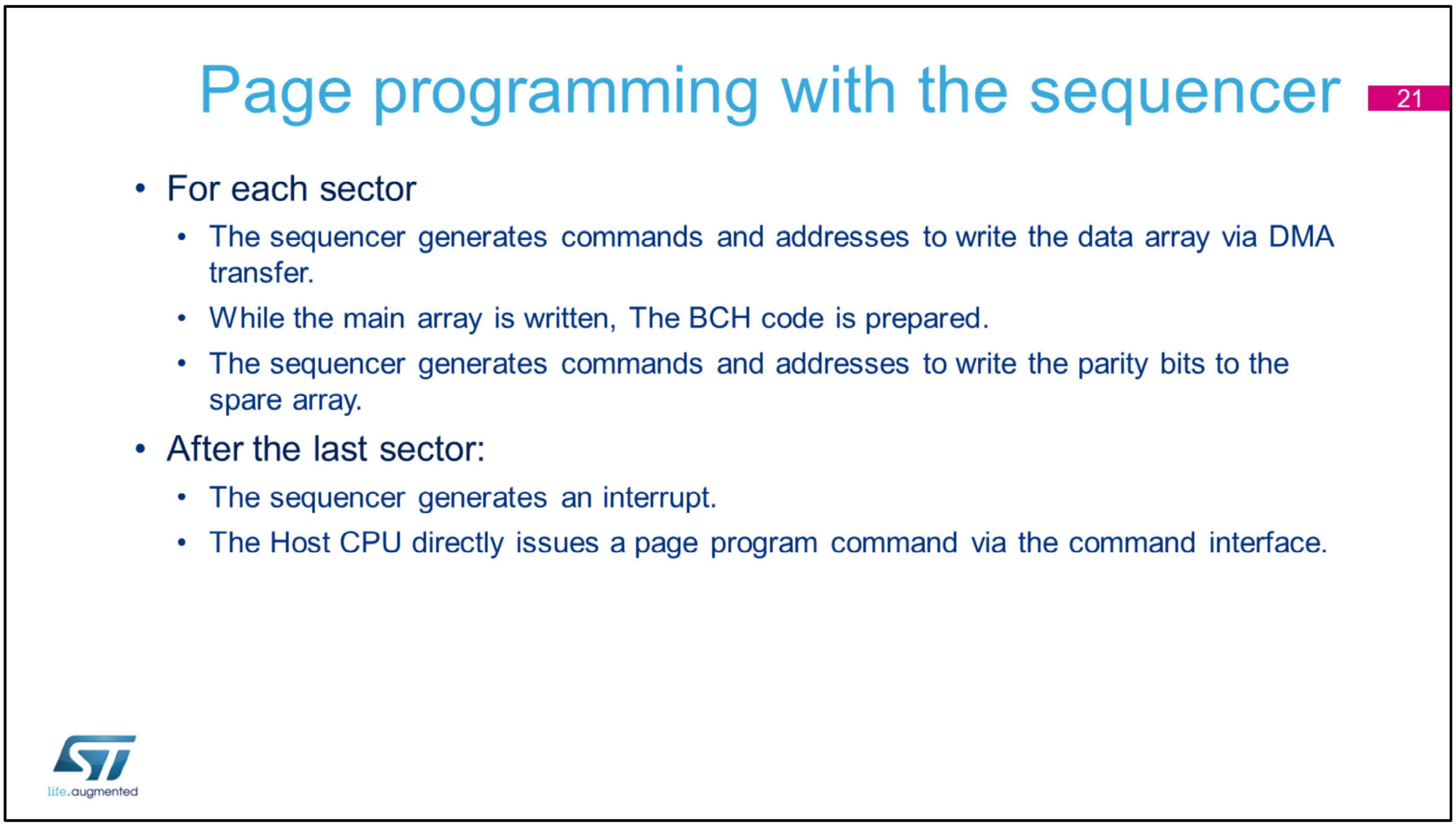


NAND COMMAND SEQUENCER

- Perform one command after another to program or read a complete page
- Support the BCH encoding/decoding, parity bits calculation and errors detection and location
- Use DMA channels for data transfer
- Fully flexible with regard to the number of sectors per page and command format

life.augmented

Data and ECC byte accesses can be managed automatically by the FMC command sequencer using DMA channels for the data transfer.
One DMA channel is required for write operations and two DMA channels for read operations.
The sequence is performed at page level, each page being one or several 512-byte sectors.
The error correction is applied to each sector with the redundant bits located in the ‘spare array’.



Page programming with the sequencer

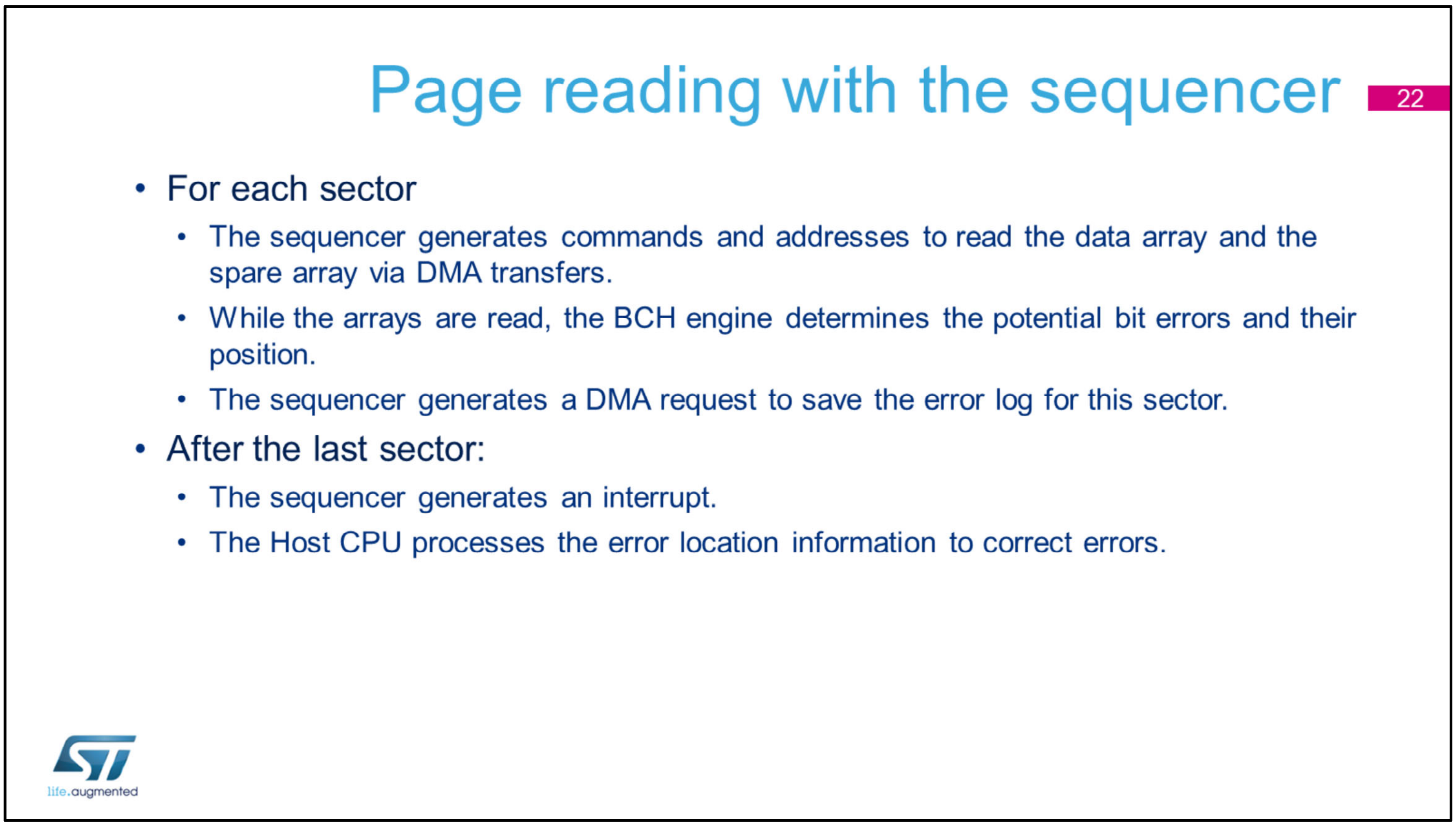
- For each sector
  - The sequencer generates commands and addresses to write the data array via DMA transfer.
  - While the main array is written, The BCH code is prepared.
  - The sequencer generates commands and addresses to write the parity bits to the spare array.
- After the last sector:
  - The sequencer generates an interrupt.
  - The Host CPU directly issues a page program command via the command interface.

The NAND Flash controller sequencer can perform one operation after another to program a complete page without any software intervention.
For each 512-byte sector in the page, the sequencer :

- Sends the command and the address to the NAND Flash main array. This requires only 1 DMA channel.
- Triggers a DMA request to write data to the NAND Flash main array.
- While data are written to the main array, the parity bits are computed by the BCH engine.
- Sends the command and the address to the NAND Flash spare array.
- Writes the parity bits to the NAND Flash spare array.

Once all sectors are written to the main and spare arrays, the sequencer generates the completion interrupt to the Host CPU to issue a page program command to the NAND Flash memory device.

The software overhead is only one interrupt per page.
During the interrupt, the next page can be programmed in the sequencer.



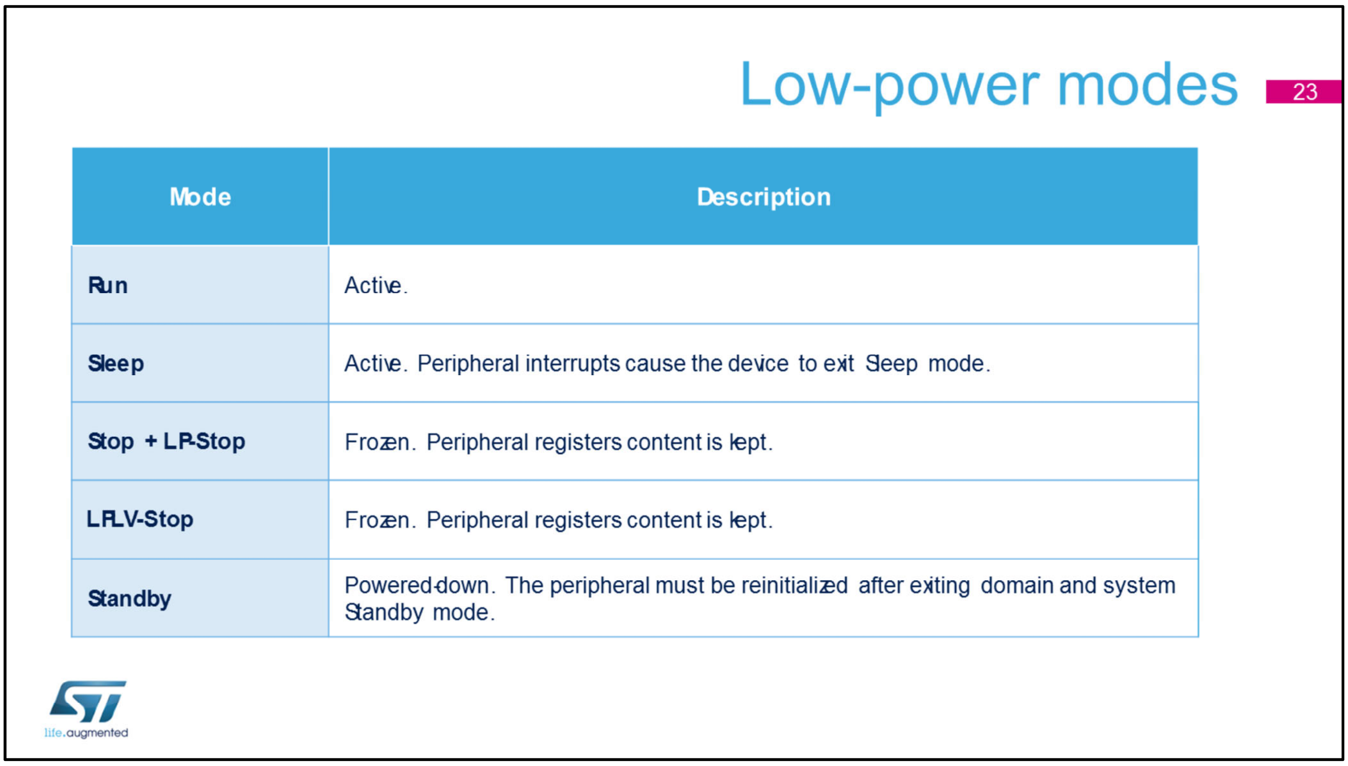
Page reading with the sequencer

- For each sector
  - The sequencer generates commands and addresses to read the data array and the spare array via DMA transfers.
  - While the arrays are read, the BCH engine determines the potential bit errors and their position.
  - The sequencer generates a DMA request to save the error log for this sector.
- After the last sector:
  - The sequencer generates an interrupt.
  - The Host CPU processes the error location information to correct errors.

The NAND Flash controller sequencer can be preprogrammed to perform all operations to read a complete page without any software intervention.
For each 512-byte sector in the page, the sequencer :

- Sends commands and addresses to read the NAND Flash main array and the associated parity bits located in the spare array. This requires 2 DMA channels.
- Triggers a DMA request to copy data from the main array to a memory buffer.
- The sequencer keeps the data read access and the DMA transfer in sync.
- While data and parity bits are read, the BCH engine computes the error location polynomial to determine the potential presence and position of bit errors. This information is then recorded in an error log.
- When the error log is ready, the sequencer generates a DMA request to save the error log for this sector.

Once all sectors are read, the sequencer generates a completion interrupt to the Host CPU to process the error log and correct the erroneous bits in the memory buffer.
The software overhead is only one interrupt per page. During the interrupt, the next page can be programmed in the sequencer.

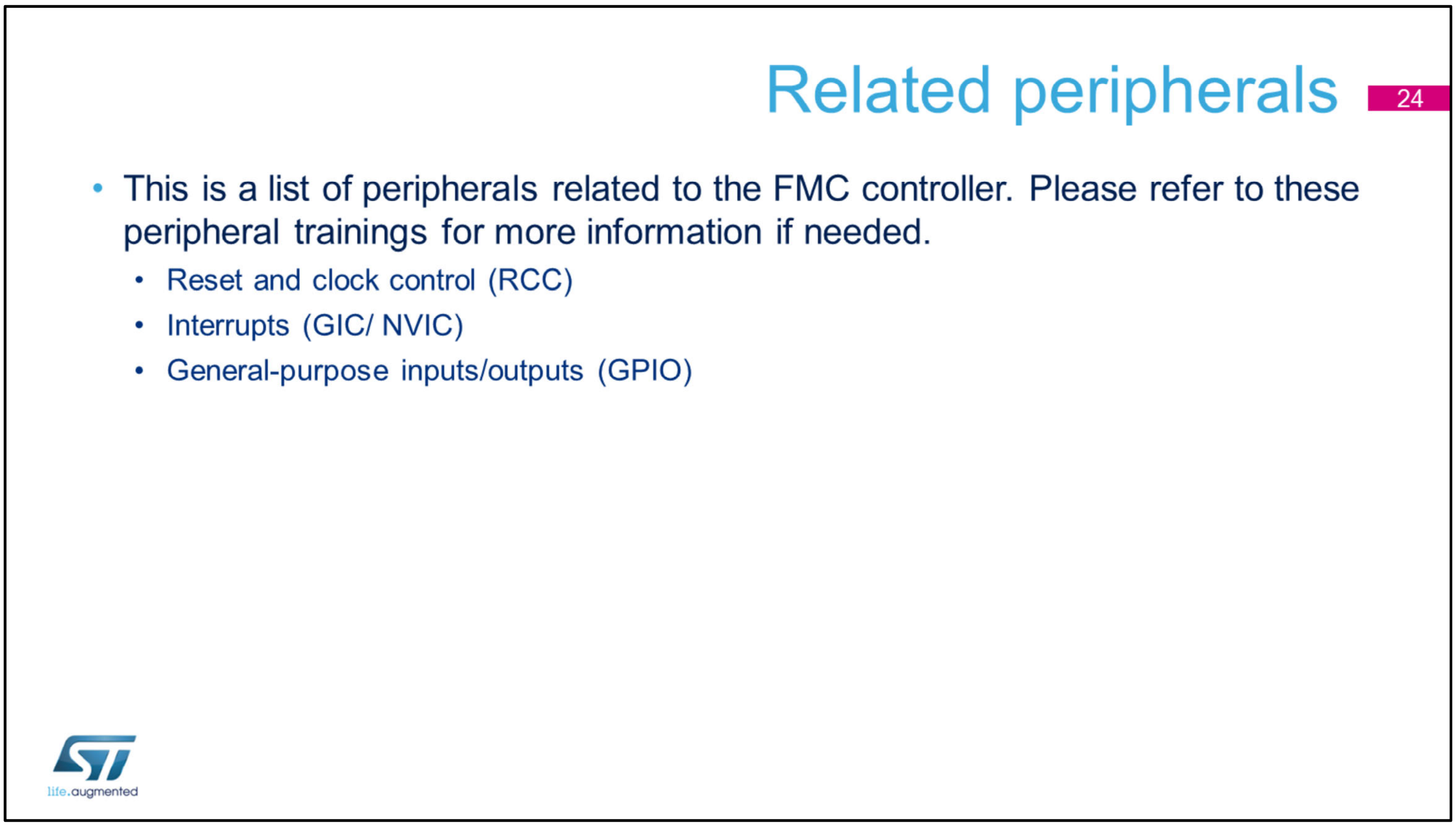


Low-power modes

Mode	Description
Run	Active.
Sleep	Active. Peripheral interrupts cause the device to exit Sleep mode.
Stop + LP-Stop	Frozen. Peripheral registers content is kept.
LPLV-Stop	Frozen. Peripheral registers content is kept.
Standby	Powered-down. The peripheral must be reinitialized after exiting domain and system Standby mode.

The FMC is active in Run and Sleep modes. A FMC interrupt can cause the device to exit Sleep mode. The device is not able to perform any communication in Stop and Standby modes. It is important to ensure that all transmissions are completed before the FMC controller is disabled or the domain or system is switched down to Stop or Standby modes.

To retain external SDRAM memory data while in Stop or Standby modes, it can be put in the Self-refresh mode prior to entering Stop or Standby modes.



Related peripherals

- This is a list of peripherals related to the FMC controller. Please refer to these peripheral trainings for more information if needed.
  - Reset and clock control (RCC)
  - Interrupts (GIC/ NVIC)
  - General-purpose inputs/outputs (GPIO)

life.augmented

Here is a list of peripherals related to the FMC interface. Users should be familiar with all the relationships between these peripherals to correctly configure and use the FMC controller.