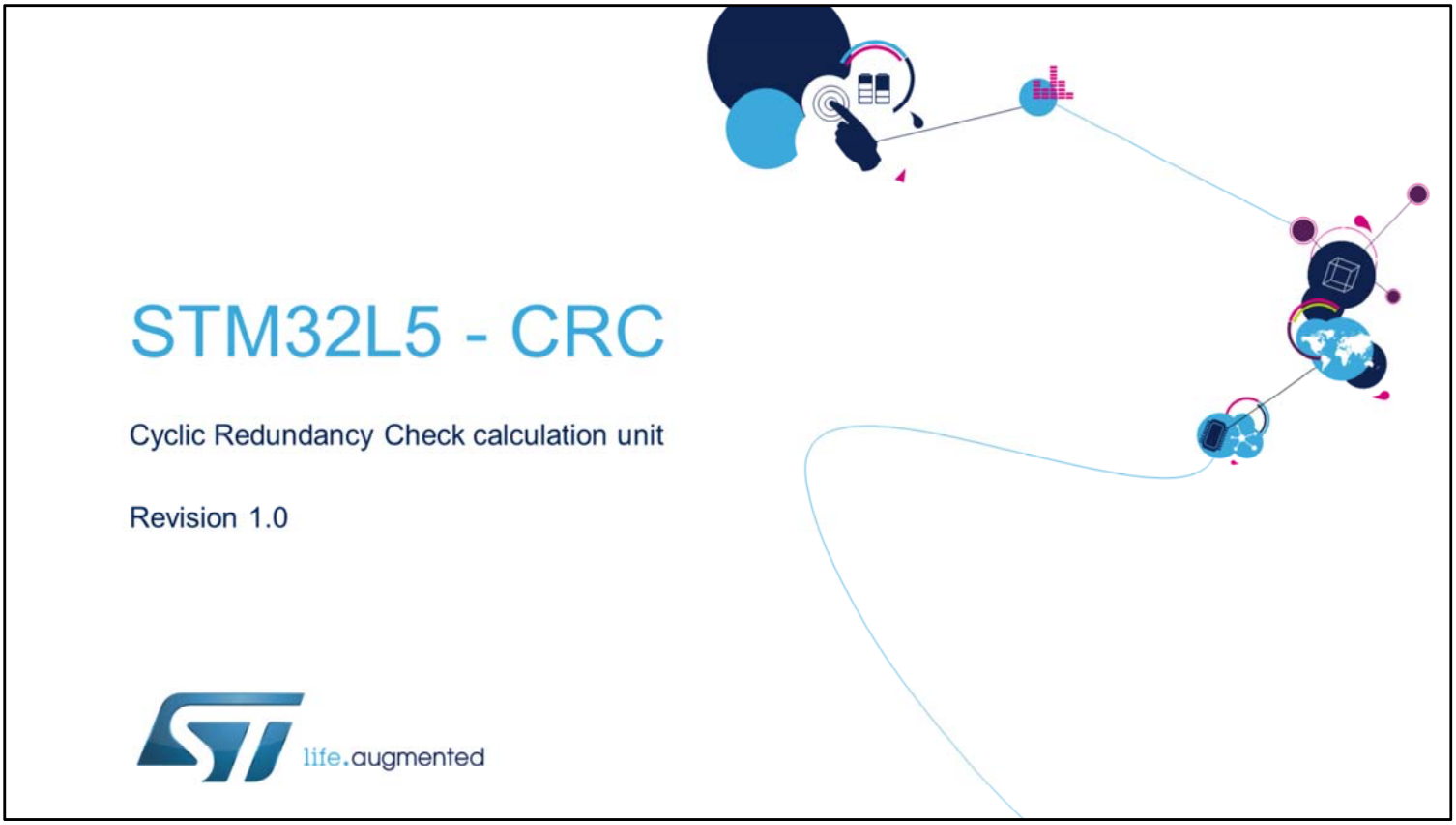
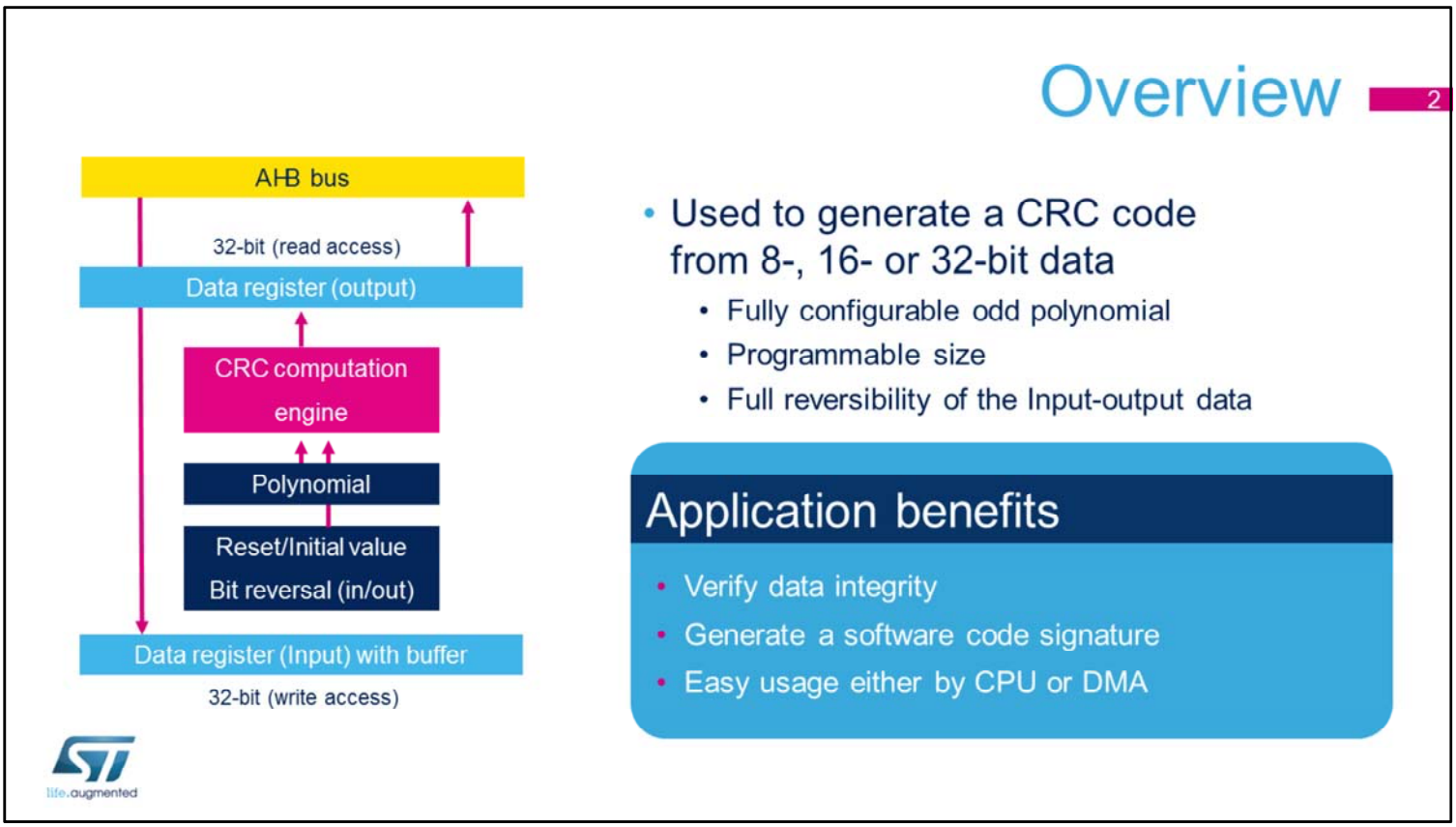


Hello, and welcome to this presentation of the STM32 Cyclic Redundancy Check calculation unit. It covers all the features of this unit, which generates a code used to detect errors.



CRC stands for Cyclic Redundancy Check. The CRC calculation unit is used to generate a CRC code on 8, 16, 32 bits of data using a configurable polynomial value and size. The CRC-based techniques can be used to verify data transmission or storage integrity.

The CRC calculation can also be used to compute the signature of the application software during runtime, to be compared with a reference signature generated at link time and stored at a given memory location.

Thanks to the full configurability of the STM32 CRC calculation unit, software overhead is reduced to a minimum. Additionally, the DMA controller can be used for back to back CRC calculations over a large block of data while the CPU is performing other tasks or in sleep mode.

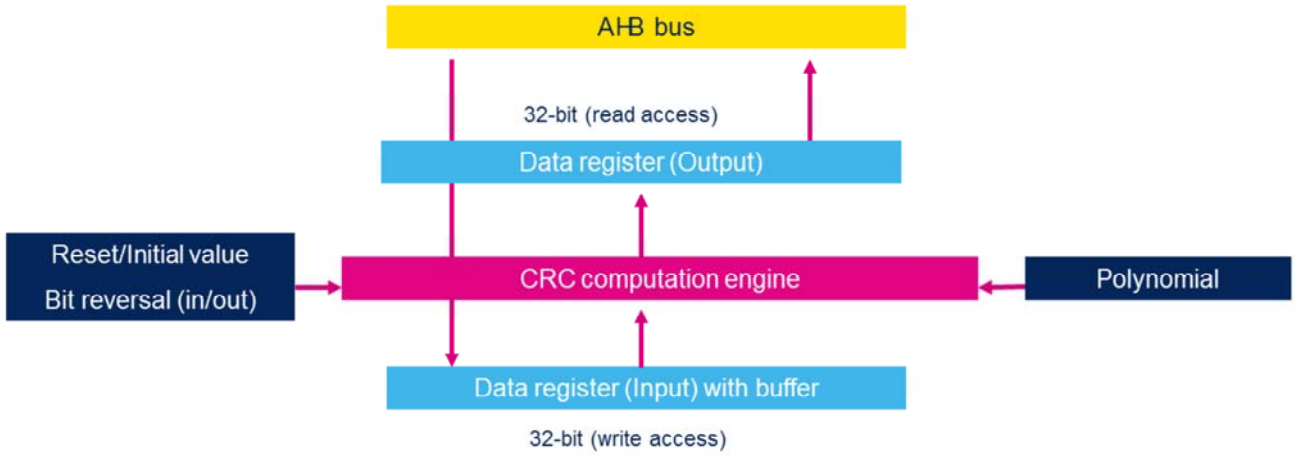
Key features

- Fully programmable odd polynomial:
  - By default uses CRC-32 (Ethernet) polynomial: 0x04C11DB7
  - Which represents polynomial $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$
- Programmable polynomial size (7, 8, 16, 32 bits)
- Programmable initial value (default = 0xFFFFFFFF)
- Various endian schemes for input and output data

life.augmented

The CRC calculation unit integrated in the STM32 features a fully programmable odd-value polynomial with a programmable size of 7, 8, 16 or 32 bits.
The initial value is also fully programmable allowing for very flexible run-time CRC code generation.
The peripheral can be configured to support the big or little endian formats of the input and output data supporting various communication protocols.

Block diagram



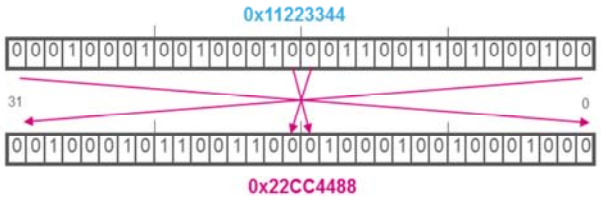
The CRC calculation unit is connected to the AHB bus interface for an optimal performance.
It contains a single 32-bit register which is used for both writing and reading by either the CPU or DMA controller.
The initial value, input/output bit reversal, and polynomial coefficients are configured by the software during the initialization phase before starting a new CRC calculation.
The CRC computation engine will hold the result of any previous CRC calculation which then becomes the initial value for the next calculation, thus allowing a new CRC computation of a data block.

Input and output format

5

Reduced software overhead

- Input data can be reversed by 8, 16, or 32 bits
  - Depending on the REV_IN[1:0] value in the CRC_CR register
  - Example, if input data 0x1A2B3C4D is used for CRC calculation:
    - 0x58D43CB2 with bit-reversal done by byte
    - 0xD458B23C with bit-reversal done by half-word
    - 0xB23CD458 with bit-reversal done on the full word
- Output data can be reversed by 32 bits
  - Setting REV_OUT bit in the CRC_CR register
  - Example on output data 0x11223344:



life.augmented

The input and output format is fully configurable to reduce the software overhead. It allows the application to manage different endian schemes.

The input data can be reversed and performed on 8, 16 or 32 bits depending on the REV_IN[1:0] bits in the CRC_CR register.
For example: input data 0x1A2B3C4D is used for CRC calculation as:

- 0x58D43CB2 with bit-reversal done by byte
- 0xD458B23C with bit-reversal done by half-word
- 0xB23CD458 with bit-reversal done on the full word

The output data can also be reversed by setting the REV_OUT bit in the CRC_CR register.

The operation is done at bit level: for example, output data 0x11223344 is converted into 0x22CC4488.

Performance

6

- One single in/out 32-bit data register for simple firmware operation
- Input buffer for successive writes without stalling the AHB bus
- The duration of the CRC computation depends on the width of input data:
  - 4 AHB clock cycles for 32-bit blocks
  - 2 AHB clock cycles for 16-bit blocks
  - 1 AHB clock cycles for 8-bit blocks
- Writes can be done by DMA while CPU is in Sleep mode
  - Low-power operation (consecutive calculations) and frees CPU for time-sensitive tasks.

life.augmented

The CRC data register includes an input buffer which allows for immediate writes by the CPU or DMA peripheral of a second data word without waiting for any wait states due to an active CRC computation.

The CRC data register can be accessed by word, right-aligned half-word, and right-aligned byte.
The duration of the computation depends on the data width:
• 4 AHB clock cycles for 32-bit blocks
• 2 AHB clock cycles for 16-bit blocks
• 1 AHB clock cycles for 8-bit blocks

The DMA controller can be used for writing to the CRC engine. This offloads the CPU for other tasks and can be used for back-to-back calculations for verifying code or data integrity.

Low-power modes

Mode	Description
Run	Active.
Low-power Run	Active.
Sleep	Active, DMA controller can be used for CRC computation.
Low-power sleep	Active, DMA controller can be used for CRC computation.
Stop	Frozen in STOP 0, STOP 1 and STOP 2 modes. Peripheral register contents are kept.
Standby	Powered-down. The CRC unit must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The CRC unit must be reinitialized after exiting Shutdown mode.

life.augmented

This slide summarizes the power modes in which the CRC calculation unit is available. The CRC calculation unit needs the main AHB clock to operate.