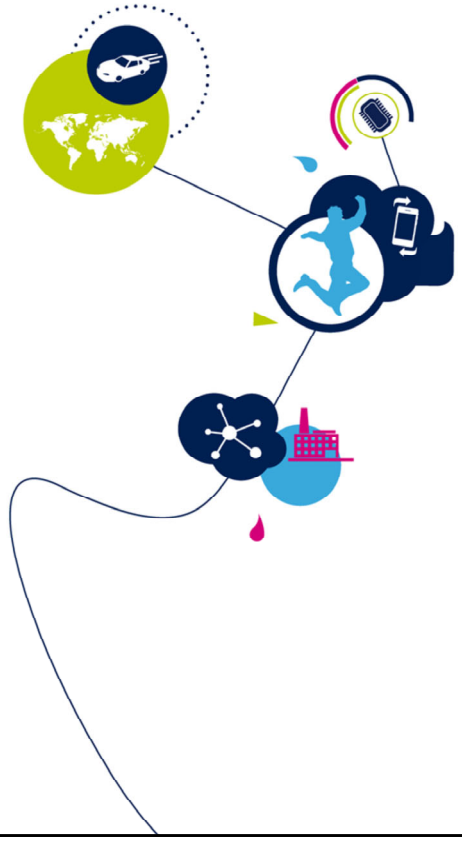
STM32WB - RNG

Random Number Generator

Revision 1.0



life.augmented

Hello, and welcome to this presentation of the STM32 Random Number Generator. The features of this peripheral, which is widely used to provide random numbers, will be covered in this presentation.

Overview

- Provides random numbers
  - Used when producing an unpredictable result is desirable.

Application benefits

- Increase the randomness of numbers
- Strongly decrease the possibility of guessing values

life.augmented

The random number generator (RNG) integrated inside STM32 products provides random numbers which are used when producing an unpredictable result is desirable. Applications can benefit from the RNG to increase the randomness of numbers or to decrease the possibility of guessing certain values.

Key features

3

- 32-bit Random Number Generator based on a noise source.
  - A set of four 32-bit random numbers can be generated at a minimum frequency of 213 clock cycles.
    - *The actual value (if higher than 213) is 16 x f_{AHB} / f_{RNG} , with a ratio of system clock versus RNG sample clock. For f_{AHB}= 32 MHz and f_{RNG}= f_{USB}/ 3 = 16 MHz, samples are available every 213 AHB cycles.*
  - Can be disabled to reduce power consumption (RNGEN=0 in RNG_CR).
- 3 flags can be triggered when:
  - DRDY: Valid random data is ready.
  - SECS: An abnormal sequence occurs on the seed (more than 64 consecutive bits at the same value “0” or “1”, or 32 consecutive patterns “01” or “10”).
  - *CECS: f_{RNG}* frequency is lower than f_{AHB} / 32 (this check can be disabled).
- 3 interrupts
  - CEIS: to indicate a clock error.
  - SEIS: to indicate a seed error.
  - DRDY: to indicate that a valid random data is ready.

life.augmented

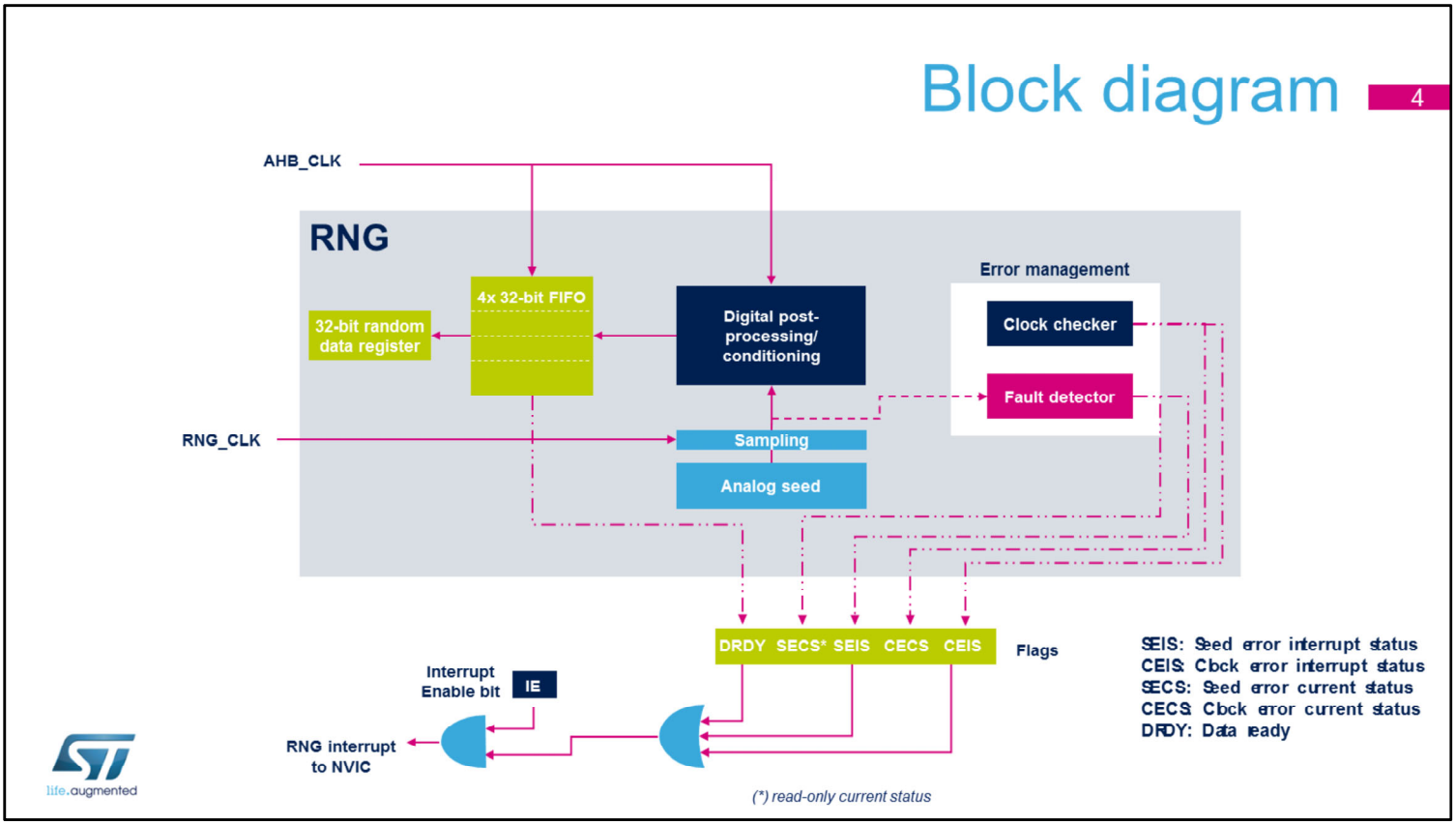
The RNG peripheral is based on continuous analog noise that provides a random 32-bit value which will be explained in detail later on. The RNG is able to generate four 32-bit random numbers at a minimum frequency of 213 system clock cycles. Rule of thumb is the lower the RNG clock, the better the entropy for the sampled random source.

The Data Ready flag is set in the status register when a set of new random data is ready and validated. It must always be used.

The RNG performs a basic verification of randomness on the provided data. For example, if more than 64 consecutive bits have the same value (0 or 1) or there are more than 32 consecutive alternating 0s and 1s, a Seed Error Current status flag is set.

A Clock Error Current status flag is set if the RNG clock is less than HCLK clock divided by 32. This check can be

disabled, especially when the RNG clock is initialized low for maximum entropy.
An interrupt source can also be enabled to indicate an abnormal seed sequence or frequency error.



This simplified block diagram of the RNG shows its basic functional and control modules.
The random number generator is based on an analog circuit made of several ring oscillators whose outputs are sampled then XORed to generate the seeds that feed a digital post-processing block that is able to produce four 32-bit random numbers per round of computation.
The sampling of analog seeds is clocked by a dedicated RNG clock signal so that the quality of the random number is independent of the HCLK frequency. The contents of the post-processing block is transferred into the data register through a four-word FIFO. The Data Ready flag (DRDY) is triggered as soon as the FIFO is full, and is automatically reset when no more data can be read back from the RNG.
In parallel, an Error Management block verifies the correct seed behavior and the frequency of the RNG

source clock.
Status bits are set and an interrupt is triggered if an abnormal sequence is detected in the seed or if the RNG frequency is too low.
The RNG frequency error check must be disabled if the RNG clock is fixed below AHB_CLK/ 32 (for example, for quality reasons).

Low-power modes

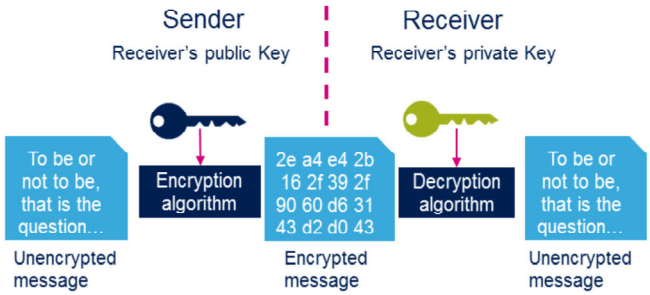
Mode	Description for RNG peripheral
Run	Active.
Sleep	Disabled in RCC or in the RNG (RNGEN=0). Keeping RNG enabled remove latency before new random sample is available, because of the RNG initialization time
Low-power run	Disabled in RCC for lowest power consumption.
Low-power sleep	
Stop 0/1/2	
Standby	Powered-down. The peripheral must be reinitialized after exiting Standby mode.
Shutdown	Powered-down. The peripheral must be reinitialized after exiting Shutdown mode.

life.augmented

The true Random Number Generator is only active in Run mode. It can be kept enabled in Sleep mode to avoid the latency at initialization time. It is disabled for the other low-power modes and is completely powered-down in Standby or Shutdown modes.

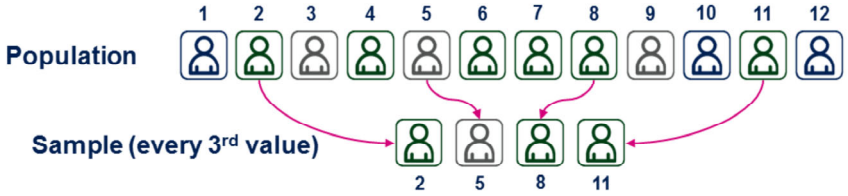
Application examples

- Cryptography



- Games

- Statistical sampling



life.augmented

The RNG can be used for a wide range of applications including cryptography, games, and statistical sampling. For example, all the security of cryptography algorithms are connected to the impossibility of guessing the key. So the key has to be a random number, otherwise the attacker can guess it.

Related peripherals

- Peripherals linked to the RNG
  - RCC (RNG clock control, RNG enable/reset)
  - Interrupts (RNG interrupt mapping)

life.augmented

This is a list of peripherals related to the random number generator. Please refer to these trainings for more information if needed.

References

- AN4230: STM32 microcontrollers random number generation validation using NIST statistical test suite.
  - AN4230 provides guidelines to verify the randomness of the numbers generated by the random number generator peripheral embedded in a selection of STM32 microcontrollers. This verification is based on the National Institute of Standards and Technology (NIST) Statistical Test Suite (STS) SP 800-22, which was published and updated as SP800-22rev1a (April 2010).
  - The NIST test suite was run on a selection of STM32 boards embedding RNG peripheral. The results are provided in the firmware folder 'NIST_Test_Suite_OutputExample'.

life.augmented

For more details, please refer to application note AN4230 about using the NIST statistical test suite to validate the random numbers generated by a selection of STM32 MCUs.