26 Random Number Generator (RNG)

26.1 Introduction

The ESP32-C6 contains a true random number generator, which generates 32-bit random numbers that can be used for cryptographical operations, among other things.

26.2 Features

The random number generator in ESP32-C6 generates true random numbers, which means random numbers generated from a physical process, rather than by means of an algorithm. No number generated within the specified range is more or less likely to appear than any other number.

26.3 Functional Description

Every 32-bit value that the system reads from the LPPERI_RNG_DATA_REG register of the random number generator is a true random number. These true random numbers are generated based on the **thermal noise** in the system and the **asynchronous clock mismatch**.

- **Thermal noise** comes from the high-speed ADC or SAR ADC or both. Whenever the high-speed ADC or SAR ADC is enabled, bit streams will be generated and fed into the random number generator through an XOR logic gate as random seeds.
- RC_FAST_CLK is an **asynchronous clock** source and it increases the RNG entropy by introducing circuit metastability.

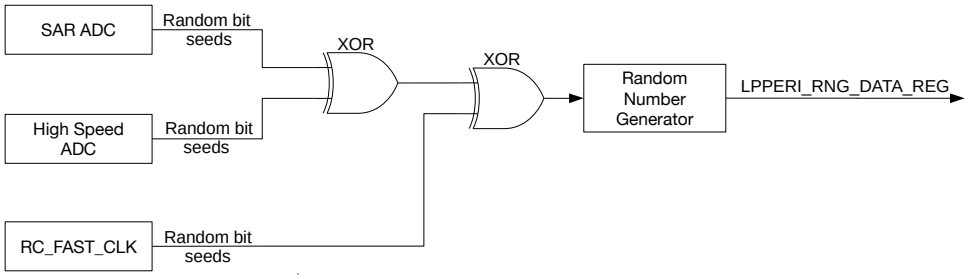


Figure 26-1. Noise Source

When there is noise coming from the SAR ADC, the random number generator is fed with a 2-bit entropy in one clock cycle of RC_FAST_CLK, which is generated from an internal RC oscillator (see Chapter 8 *Reset and Clock* for details). Thus, it is advisable to read the LPPERI_RNG_DATA_REG register at a maximum rate of 1 MHz to obtain the maximum entropy.

When there is noise coming from the high-speed ADC, the random number generator is fed with a 2-bit entropy in one APB clock cycle, which is normally 80 MHz. Thus, it is advisable to read the LPPERI_RNG_DATA_REG register at a maximum rate of 5 MHz to obtain the maximum entropy.

A data sample of 2 GB, which is read from the random number generator at a rate of 5 MHz with only the high-speed ADC being enabled, has been tested using the Dieharder Random Number Testsuite (version 3.31.1). The sample passed all tests.

26.4 Programming Procedure

When using the random number generator, make sure at least either the SAR ADC, high-speed ADC[1], or RC_FAST_CLK[2] is enabled. Otherwise, pseudo-random numbers will be returned.

- SAR ADC can be enabled by using the DIG ADC controller. For details, please refer to Chapter 39 *On-Chip Sensor and Analog Signal Processing*.
- High-speed ADC is enabled automatically when the Wi-Fi or Bluetooth module is enabled.
- RC_FAST_CLK is enabled by setting the RTC_CNTL_DIG_FOSC_EN bit in the RTC_CNTL_CLK_CONF_REG register.

Note:

1. Note that, when the Wi-Fi module is enabled, the value read from the high-speed ADC can be saturated in some extreme cases, which lowers the entropy. Thus, it is advisable to also enable the SAR ADC as the noise source for the random number generator for such cases.
2. Enabling RC_FAST_CLK increases the RNG entropy. However, to ensure maximum entropy, it's recommended to always enable an ADC source as well.

When using the random number generator, read the LPPERI_RNG_DATA_REG register multiple times until sufficient random numbers have been generated. Ensure the rate at which the register is read does not exceed the frequencies described in section 26.3 above.

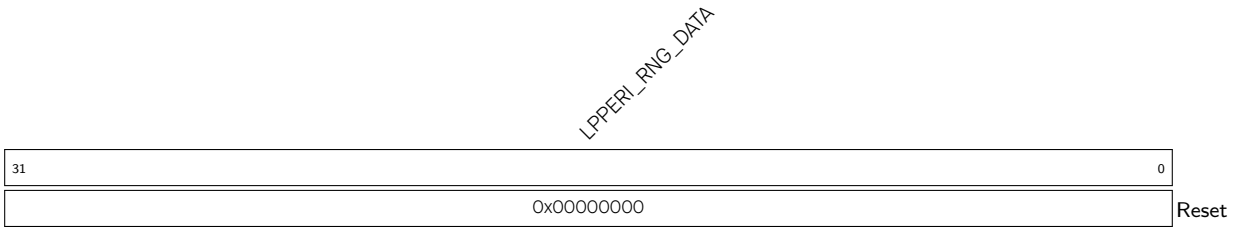
26.5 Register Summary

The abbreviations given in Column **Access** are explained in Section *Access Types for Registers*.

Name	Description	Address	Access
LPPERI_RNG_DATA_REG	Random number data	0x600B_2808	RO

26.6 Register

Register 26.1. LPPERI_RNG_DATA_REG (0x600B_2808)



LPPERI_RNG_DATA Random number source. (RO)