SP 800-90B Non-Proprietary Public Use Document
Infineon SLC37 32-bit Security Controller V11 Entropy Source
Document Version 1.1

Hardware Versions:

SLC37 32-bit Security Controller V11

Infineon Technologies AG
Am Campeon 1-15
Neubiberg, BY 85579, Germany

April 6, 2023

Revision History

Version	Change
April 6, 2023, v1.0	Initial release
January 16, 2023, v1.1	Updated entropy rate

Table of Contents

Description

The Infineon SLC37 32-bit Security Controller V11 Entropy Source is a physical entropy source. It is a Non-IID source and was tested for the platform listed on the title page. The configuration is set by register reset values hard coded in the design. No modification on the configuration is performed by firmware.

Security Boundary

The Infineon SLC37 32-bit Security Controller V11 Entropy Source is entirely contained within a single-chip module.

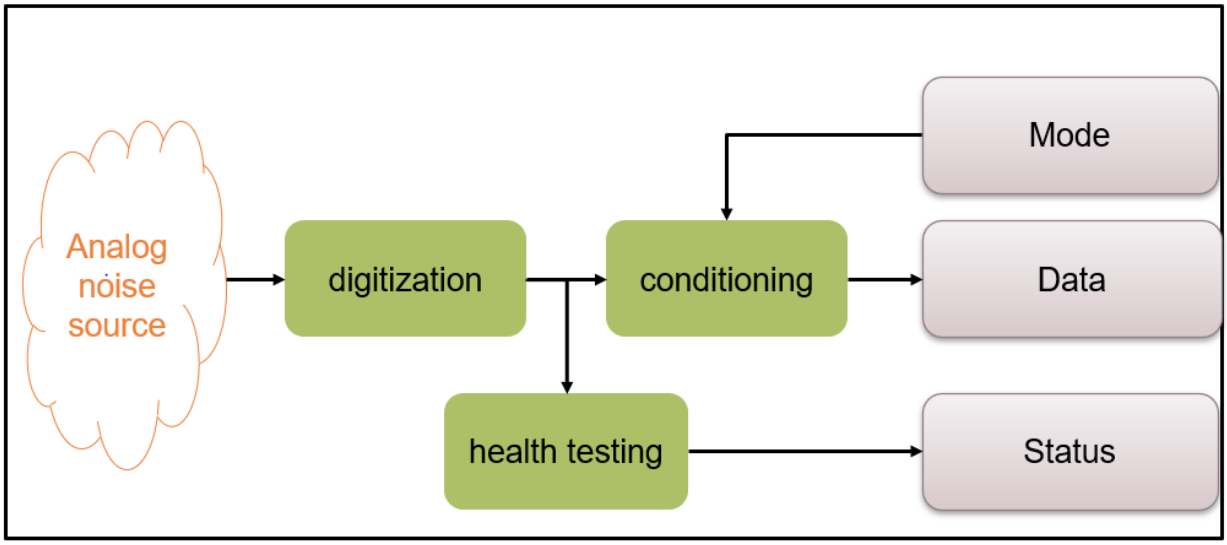

Figure 1. Infineon SLC37 32-bit Security Controller V11 Entropy Source

Operating Conditions

The following table lists the relevant operating conditions for which the Infineon SLC37 32-bit Security Controller V11 Entropy Source was characterized.

Parameter	Value	Description
Temperature	-40°C to 115°C	Operating ambient temperature range
VCC	1.62 V to 5.50 V	Supply Voltage

Configuration Settings

To select the SP 800-90B validated source the Infineon SLC37 32-bit Security Controller V11 Entropy Source must be set to **TRNG Mode**.

To start the entropy source, the mode must be set from **Configuration Mode** into **Operational Mode**. There are no further configuration settings available to the operator.

Physical Security Mechanisms

The Infineon SLC37 32-bit Security Controller V11 Entropy Source is part of a single-chip module that meets commercial-grade specifications for power, temperature, reliability, and shock/vibrations. The module employs standard passivation techniques.

Conceptual Interfaces

GetEntropy interface is a read operation on the ***Data*** register of the Infineon SLC37 32-bit Security Controller V11 Entropy Source. Once the ready flag is set in the ***Status*** register, 32 bit of entropy data can be read from the ***Data*** register.
There is no dedicated *HealthTest* interface. Data is only released once the health testing is passed. In case health testing fails, a flag is set in the ***Status*** register. To restart the entropy source the ***Mode*** must be set to **Operational Mode**.
GetNoise is an internal interface only available in a dedicated privileged mode.

Min-Entropy Rate

H_submitter = 0.45 bits per symbol for a symbol size of four bits after digitization.

We claim an output (H_out) of at least 0.418 bits per bit, or 13.376 bits of min entropy per 32-bit block.

Health Tests

The RAW data after digitization is tested by the Adaptive Proportion Test (APT) and the Repetition Count Test (RCT). For the RCT, the selected cutoff is C=64 and for the APT, the selected cutoff is C=451.
Continuous health testing is performed for each generated output symbol during normal operation. An output symbol is only released to the ***Data*** register if continuous health testing was passed.
Start-up health testing is performed by applying the implemented health tests to 1024 RAW samples. This is initiated by reading 8 times 32-bit of conditioned data from the Infineon SLC37 32-bit Security Controller V11 Entropy Source ***Data*** register. If no alarm was indicated in the ***Status*** register after 8 32-bit words were read, the start-up health testing result is “pass” for the 1024 tested RAW symbols.
On demand health testing is possible by reading 8 times 32-bits of entropy data from the Infineon SLC37 32-bit Security Controller V11 Entropy Source ***Data*** register and monitoring the health test error bit in the ***Status*** register. In case of an indicated failure none of the data already retrieved during on demand health testing must be used.

Maintenance

There is no special maintenance necessary to operate the Infineon SLC37 32-bit Security Controller V11 Entropy Source.

Required Testing

There is no additional testing required by the target platform. Since *GetNoise* is an internal interface only available in a dedicated privileged mode, the user must rely on the health tests to ensure the entropy source is configured correctly and is working as expected.