Nvidia Corporation

Nvidia Random Number Generator

Version: 1.0

SP 800-90B Non-Proprietary Public Use Document

Document Version: 0.7

Prepared for:

Nvidia Corporation
2788 San Tomas Expressway
Santa Clara, CA 95051
United States of America

Phone: +1 408 486-2000
www.nvidia.com

Prepared by:

Corsec Security, Inc.
12600 Fair Lakes Circle, Suite 210
Fairfax, VA 22033
United States of America

Phone: +1 703 267 6050
www.corsec.com

Revision History

Document Version	Version	Modification Date	Modified By	Description of Changes
0.2	1.0	2023-11-30	Steele Myrick	Initial draft.
0.3	1.0	2024-02-05	Steele Myrick	Updated the PUD: Added condensing rate explanation.
0.4	1.0	2024-03-19	Steele Myrick	Updated entropy samples.
0.5	1.0	2024-04-19	Steele Myrick	Updated min-entropy and explained H value for health tests.
0.6	1.0	2026-01-21	Mira Eisenhauer	Updated to add Networking Adapter Security operating environment information.
0.7	1.0	2026-03-13	Yvonne Cliff	Updates to meet the PUD template v1.2 requirements

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

Table of Contents

1. Description 4
2. Security Boundary 5
3. Operating Conditions 6
4. Configuration Settings 7
5. Physical Security Mechanisms 8
6. Conceptual Interfaces 9
7. Min-Entropy Rate 10
8. Health Tests 11
9. Maintenance 12
10. User Verification 13
10.1 Raw Data Analysis 13
10.2 Conditioned Data Analysis 13
10.3 Restart Data Analysis 13

List of Tables

Table 1 – Nvidia Security Engines 4
Table 2 - Standards Applied 4
Table 3 – Operating Conditions 6
Table 4 – Configuration Settings 7
Table 5 – Final Raw Min-Entropy Estimates 13
Table 6 – Final Restart Min-Entropy Estimates 14

List of Figures

Figure 1 – NVRNG Entropy Source 5

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

1. Description

The Nvidia Random Number Generator (NVRNG) version 1.0 is a physical (P) entropy source that uses ring oscillators that meet the standards of NIST SP 800-90B. The entropy source consists of hardware that executes in a non-modifiable operational environment. There is no firmware. The embodiment is single-chip.

An IID claim is not made and the NVRNG meets all the requirements for non-IID compliance. The report is for the NVRNG version 1.0. The table summarizes the Nvidia Corporation Nvidia Random Number Generator Foundation Security Processor (FSP), Security Processor (SECHUB), and Platform Security Controller (PSC-SE) engines.

Table 1 – Nvidia Security Engines

Engine	Chip
FSP	Nvidia H100 Tensor Core
SECHUB	Nvidia H100 Tensor Core
PSC-SE	Nvidia Network Adapter Security

Standards applied are shown in Table 2.

Table 2 - Standards Applied

NIST Document	Title	Date
SP 800-90B	Recommendation for Entropy Sources Used for Random Bit Generation	Jan 2018
Implementation Guidance	Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program IG 9.3.A last modified July 26, 2024 IG D.J last modified November 5, 2021 IG D.K last modified March 17, 2023 IG D.O last modified May 4, 2021	Feb 2026

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

2. Security Boundary

The boundary of the NVRNG implementation is shown in Figure 1. An important contributor to the variation of jitter in a ring oscillator (RO) inverter loop is Johnson-Nyquist thermal noise, which occurs spontaneously in any conductor (regardless of quality) as a result of the thermal agitation of free electrons. Jitter is a macroscopic manifestation of this quantum-level Brownian effect. This section describes the security boundary for Network Adapter Security (PSC-SE) and H100 (FSP and SECHUB).

At the macroscopic level, we see timing jitter as the variation of a fast oscillator signal's frequency. In our case, the variation in time is within the ROs.

The ROs of the Multiple Ring Oscillators (MURO) have a stable slow clock signal that samples an unstable fast oscillator signal to generate random bits. As the sampling interval increases, the jitter of the fast oscillator signal accumulates. The foundation of generating random bits is the unpredictability of the number of fast signal periods (more precisely, half-periods) in the duration of a single slow sample clock period. The NVRNG has a conceptual entropy source (NVRNG) Security Boundary. The NVRNG entropy source contains:

- SP800-90B noise source – MURO.
- SP800-90B vetted conditioning component – AES-128-CBC-MAC.
- SP800-90B health test logic – Applied to the MURO output.

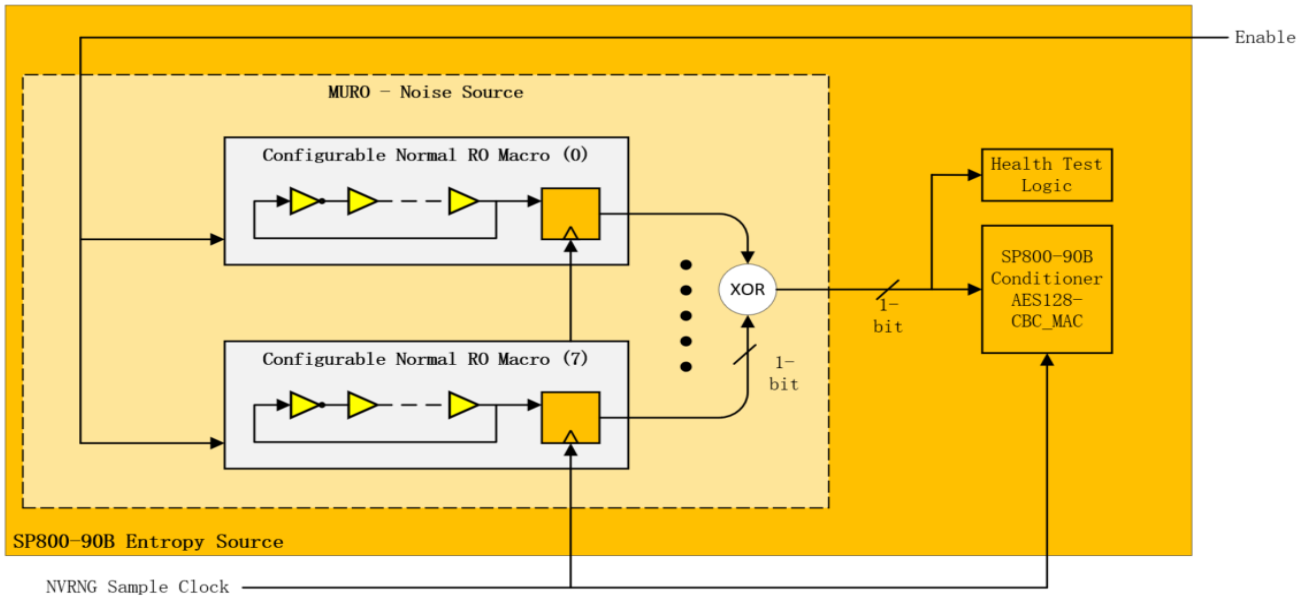

Figure 1 – NVRNG Entropy Source

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

3. Operating Conditions

Table 3 summarizes the operating conditions for each of the tested Engines within the chip.

Table 3 – Operating Conditions

Engine	Chip Clock Speed	Chip	Chip RO NVRNG Temperature Range	Chip RO Supply Voltage (DC)
FSP	540 MHz	Nvidia H100 Tensor Core	$0°C < T < 105°C$	$0.55 < V < 1.16$
SECHUB	540 MHz	Nvidia H100 Tensor Core	$0°C < T < 105°C$	$0.55 < V < 1.16$
PSC-SE	600 MHz	Nvidia Network Adapter Security	$0°C < T < 105°C$	$0.55 < V < 1.16$

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

4. Configuration Settings

The "condensing rate" is a multiplier to 128 bits that determine the total number of the noise source's output bits being input into the vetted conditioning component.

The conditioning component is a crucial part of the NVRNG. It processes the raw entropy collected from the MURO, which contains the noise source. The raw data is condensed into a form with higher entropy per bit, meaning the output is closer to being truly random and unpredictable. Table 3 summarizes the configuration setting used by the NVRNG.

The B_CTRL0.COND_BLK_NUM register field is available for the programmable condensing rate. The hardware default and validated value is 9. The conditioning component has a raw noise input of n = (B_CTRL0.COND_BLK_NUM + 1) x 4 blocks of 128 bit. For the hardware default value, there are 40 input blocks, and AES-CBC-MAC condenses these into a single output block. This process results in a 40:1 compression ratio.

Table 4 – Configuration Settings

Setting	Value
Condensing Rate	40

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

5. Physical Security Mechanisms

The NVRNG noise source is implemented in hardware only and operates within a conceptual security boundary. The ROs and digitization occur in a non-modifiable operational environment. Administrative software is outside the boundary of the entropy source and is unable to read internal oscillator data, sampling values, or raw noise directly. Raw noise source outputs are not readable during normal operation, and unconditioned raw noise is only accessible in validation mode. Additionally, ROs with physical separation are less susceptible to external interference. All the ROs are tuned relatively prime (coprime) in period as mitigation for several attacks like frequency injection. The entropy source is also monitored through health testing for abnormal behavior, which may be an indication of manipulation.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

6. Conceptual Interfaces

The below commands are available via the test command interface (T_CFG/CMD/STAT/IDC/IDD registers). The entire MURO feeds into an XOR gate, which combines the sampled bits from the MURO into a single bit. The formatted digitized binary bitstream is the raw noise source output. The noise source is selected among the NVRNG instances (FSP or SECHUB for H100, PSC-SE for Network Adapter Security) before issuing the GET_NOISE or GET_ENTROPY commands.

Raw noise data is collected from the MURO noise source's output prior to conditioning or post-processing. The GET_NOISE interface is used to collect sequential and restart data from MURO.

For GET_NOISE of 1024-bit for "restart" dataset:

- Select the right Noise Source.
- Prepare T_CMD.CMD = GET_NOISE, T_CMD.ARG = 0 (a single 1024-bit block).
- The MURO are triggered, perform their startup health tests, and collect 1024-bit can be read from DFIFO.

Each restart cycle generates an independent 1024-bit block. For each restart cycle, the process detailed above is repeated. In between cycles, the MURO instance is disabled and enabled. The output is a 1024-bit block per restart cycle read from the DFIFO.

For GET_NOISE of more bits for "continuous" dataset:

- Select the right Noise Source.
- Prepare T_CMD.CMD = GET_NOISE, T_CMD.ARG = n (n>0) to collect n x 128-bit data.
- The collected bits can be read via DFIFO. This usually requires the downstream to consume the collected bits fast enough in order to prevent stalling the MURO.

The MURO stays activated instead of being disabled/enabled, as in the restart cycles. The final output is multiples of 128-bit blocks read from the DFIFO.

The GET_ENTROPY interface is used to collect entropic data from the output of the conditioning component, which is also the output of the NVRNG. The collection interfaces are only available in NVRNG validation mode. Therefore, normal operation never has exposure of the raw noise output and the MURO's jitter is not affected.

For GET_ENTROPY:

- Select the right Noise Source.
- Prepare T_CMD.CMD = GET_ENTROPY, it activates the noise source and activates the conditioning logic to work at the configured condensing rate and generates one 128-bit seed into the SFIFO.
- The seeds from Entropy Source can be read from SFIFO.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

7. Min-Entropy Rate

A request for 128 bits of entropy from the NVRNG results in 128 bits of full entropy from the conditioning component.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

8. Health Tests

Per the SP 800-90B requirements, health tests run when the NVRNG starts, and are then run continuously while it is operating. All tests check for persistent failures based on their respective "cutoff" values, which represent expected error thresholds.

The NVRNG implements two types of health tests:

- Repetition Count Test (RCT)
- Adaptive Proportion Test (APT)

The RCT and APT tests are implemented as described in SP 800-90B.

The NVRNG does not differentiate between intermediate and permanent failures. It logs the error for the administrative SW to handle. If the noise source deteriorates to a level, determined by health tests, the NVRNG will report catastrophic error very frequently or it will not be usable. It is up to the client to report as fatal. The NVRNG entropy source stops outputting entropic bits if any health test error is detected.

For the APT cutoff, the formula provided from SP800-90B Section 4.4.2 is used with a value of $H = 0.1520$. This is the $H_{submitter}$ value for the H100 OEs. For the Network Adaptor Security (PSC-SE) OE, $H_{submitter}$ is slightly lower at 0.146, but as this is a worst-case estimate and actual min-entropy is expected to be higher, using health tests calculated for the higher min-entropy is not a problem.

The entropy source is expected to produce raw data with min-entropy of at least 0.1520 bits per bit (corresponding to $\Pr(0)$ and $\Pr(1) \leq 0.9000$), and it provides full-entropy output provided the raw data has min-entropy $\geq$ 0.0375 bits/bit, corresponding to $\Pr(0)$ and $\Pr(1) \leq 0.9743$. The analysis script evaluates MURO health-test behavior across a range of XOR bias probabilities by running over a million trials per round. It shows that for $\Pr(0)$ or $\Pr(1)$ values between approximately 0.84 and 0.93, the tests almost never fail, indicating a low true failure probability. Failures first become noticeable around $\Pr(0)$ or $\Pr(1)$ ≈ 0.933 and increase gradually, then sharply near 0.955.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

9. Maintenance

There are no maintenance requirements for the NVRNG.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

10. User Verification

Raw data can be collected by the user by using the GET_NOISE interface as described in Section 6. Through the interface 1,000,000 consecutive samples of raw data should be collected. The user should also collect 1,000,000 samples of raw data in groups of 1,000 samples where the NVRNG is restarted after each group of data for the restart tests. The user should then ensure that the NIST non-IID test suite estimates the min-entropy to be at least the $H_{submitter}$ values shown below in Section 10.3. Results of the testing carried out by the vendor are shown in the following subsections for reference. The user's results should be similar.

10.1 Raw Data Analysis

The NVRNG entropy scheme does not make an IID[1] claim, so the NIST non-IID test suite was run on the raw (unconditioned) entropy data. The 1,000,000 consecutive raw data samples were collected directly from the output of the XOR-gate in the MURO. This data is not output directly to the user but is accessible while in Validation mode.

Table 5 – Final Raw Min-Entropy Estimates

Engine	Entropy Result	MURO Output Entropy
H100 (FSP)	H_original	0.771944
H100 (SECHUB)	H_original	0.865178
Network Adapter Security (PSC-SE)	H_original	0.889691

10.2 Conditioned Data Analysis

NVRNG uses an AES-128-CBC-MAC as a conditioning component. Since it is a vetted conditioner, no conditioned data needs to be tested.

10.3 Restart Data Analysis

For the restart tests, the raw entropy data is collected for 1,000 MURO instances allocated sequentially. That means, for one collection of raw entropy, one MURO instance is allocated. After the conclusion of the data gathering, the MURO instance is deallocated and a new MURO instance is allocated for the next restart test round. This process continues until there is a 1,000 by 1,000 binary matrix.

The following data was collected directly from the output of the XOR-gate in the MURO. This data is not output directly to the user but is accessible while in Validation mode.

1 IID – Independent and identically distributed

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.

Table 6 – Final Restart Min-Entropy Estimates

Entropy Result	H100 (FSP) XOR Output Entropy	H100 (SECHUB) XOR Output Entropy	Network Adapter Security (PSC-SE) XOR Output Entropy
H_r	0.863437	0.864019	0.857026
H_c	0.768092	0.863607	0.821809
H_I	0.1520	0.1520	0.146
min(H_r, H_c, H_I):	0.1520	0.1520	0.146

The restart tests define the min-entropy as the minimum of the row min-entropy, the column min-entropy, and the estimated $H_{submitter}$. The min-entropy for the of the NVRNG is 0.1520 for the H100 OEs and 0.146 for the Network Adapter Security OE (since these are the estimated $H_{submitter}$ values). The min-entropy for the rows and the min-entropy for the columns were both close to the values calculated for the raw noise data, showing that the restart data is not degraded in any way.

©2026 Nvidia Corporation
This document may be freely reproduced and distributed whole and intact including this copyright notice.