

Digital Random Number Generator (DRNG) SP800-90B Non-Proprietary Public Use Document (PUD) for Intel® Xeon® 6700-Series Processor with E-Cores (Sierra Forest)

Entropy Source RNG_ES_4.2

September 2026

Revision 0.2.0

Document Number: 903034



Contents

1 Description 5

2 Security Boundary 6

3 Operating Conditions 8

4 Configuration Settings 9

5 Physical Security Mechanisms 10

6 Conceptual Interfaces 11

7 Minimum Entropy Rate 11

8 Health Tests 14

9 Maintenance 15

10 Required Testing 16

Appendix A Processors 17

Figures

Figure 2-1. DRBG Security Boundary 7

Figure 7-1. Entropy Levels on Noise Source and Conditioning Chain 12

Tables

Table 1-1. List of Applicable Devices 5

Table 3-1. Operating Conditions 8

Revision History

Revision Number	Description	Date
0.1.0	• Initial release of the document.	March 2026
0.2.0	• Updated Chapter 2 • Updated Chapter 6: Opcode and operand sizes • Updated Chapter 10	September 2026

1 Description

The Intel Digital Random Number Generator entropy source is a physical (P) entropy source.

The circuit component is RNG_ES_3.

The synthesizable RTL component is IP-DRNG4.2_PIC9.

All components of the entropy source are hardware, consisting of logical and electronic components bound in a rectangle of silicon. There is no firmware or software within the entropy source.

The entropic data from the entropy source is Non-Independent and Identically Distributed (IID).

This entropy source has been evaluated in accordance to NIST SP 800-90B (January 2018) and Implementation Guidance for FIPS 140-3 and the Cryptographic Module Validation Program (April 18, 2025).

Cryptographic module testing was performed on a system equipped with an Intel® Xeon® 6556P-B processor, part of the Intel® Xeon® 6700-Series processor with E-Cores (codename Sierra Forest), implemented on the Sierra Forest core die, and packaged in an FCBGA4368 format.

This Public Use Document (PUD) is applicable to the following products:

Table 1-1. List of Applicable Devices

Processor Family Name
Intel® Xeon® 6700-series processor with E-Cores (Sierra Forest): 67xxE SKU Series

A complete list of devices corresponding to each CPU family listed in Table 1-1 is provided in Appendix A Processors.

For guidance on identifying the processor in a system, refer to Intel's official support documentation available at:

- https://www.intel.com/content/www/us/en/support/articles/000006059/processors.html
- https://www.intel.com/content/www/us/en/support/articles/000006831/processors/intel-processor.html
- https://www.intel.com/content/www/us/en/support/articles/000059657/processors/intel-xeon-processors.html

2 Security Boundary

The entropy source security boundary surrounds the set of components referred to as the Digital Random Number Generator (DRNG) core, which includes the noise source, digitizer, Continuous Health Tests (CHT), Advanced Encryption Standard (AES) - Chassis Bridge Controller (CBC) - Message Authentication Code (MAC) Vetted Conditioning Component, Deterministic Random Bit Generator (DRBG), Non-deterministic Random Bit Generator (NRBG), and register interface.

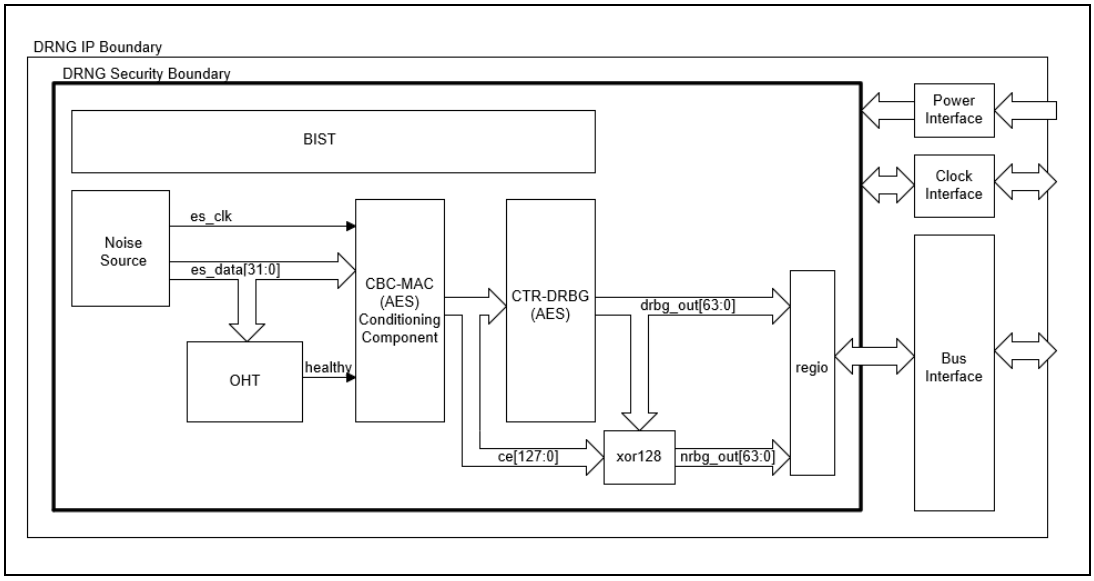
For use as a full entropy source, the NRBG output of the DRNG is the necessary output. This is accessed through the Intel CPU instruction "`RdSeed`".

The security boundary is a sub-boundary within the DRNG. Components outside the security boundary but within the DRNG are to attach to the chip's local bus, clock, and power systems.

The bold outline in Figure 2-1 shows the security boundary of the entropy source and other RNG components.

The width of these operations is 128 bits, driven by the output size of AES. The register sizes are 64 bits, and in the logic, this width transforming is performed using a First In, First Out (FIFO) that is 64 bits wide and takes in 128 bits as two 64-bit entries and outputs 64 bits to match the register width.

Figure 2-1. DRBG Security Boundary



In the context of an Intel CPU, the DRNG register that presents the NRBG output to the local bus is called `egetdata`. This register is read by the CPU whenever the `RdSeed` instruction is executed. The result is passed to the target register of the instruction and the success or failure signaled in the carry flag.

This report is applicable to the devices in Table 1-1. The silicon manufacturing process, Integrated Circuit (IC) design, and DRNG design are identical for each of these devices. Differences between these devices are the set of enabled features, which has no influence on the RNG behavior.

3 Operating Conditions

The entropy source is guaranteed to operate within the designated operating envelope in the datasheet of the chips. In the Intel® Xeon® processors listed in Table 1-1, the operating envelope is as follows.

Table 3-1. Operating Conditions

Parameter	Minimum	Maximum
Temperature	-10°C	100°C

4 Configuration Settings

There are no configuration settings accessible at any privilege level to software running on the CPU.

5 Physical Security Mechanisms

The RNG contains a security boundary described in chapter 2 Security Boundary. In the operating mode of the RNG, there is hardware enforcement of the security of the boundary.

Specifically:

- Diagnostic output of raw data from the boundary is disabled.
- Debugging port access to the internal state of the RNG is disabled at the boundary.
- Configuration input written to registers is ignored, and the default configuration is enforced.
- Algorithms are implemented to detect stuck output failures.
- Register write privileges are enforced in hardware.

When an alarm is triggered, the RNG resets itself and re-runs Built-In Self-Test (BIST). It is not possible to distinguish between a failure due to an attack, or environmental bounds violation, or a rare false positive error. The re-running of BIST will lead to the RNG failing if the BIST fails. In the case of a transitory error, the RNG will recover when BIST is re-run.

The packaging of the chip is a tamper-evident enclosure.

6 Conceptual Interfaces

For consuming applications running in software on the CPU, `GetEntropy(n)` is implemented by the `RdSeed` instruction, where *n* can be one of 16, 32, or 64, depending on the size of the target register.

Internally to the security boundary, for feeding conditioner output data to the DRBG, `Get_entropy()` is implemented in digital logic between the conditioner and DRBG in the form of a 128-bit wide First In, First Out (FIFO).

RDSEED retrieves a hardware generated random seed value from the SP800-90B and C compliant ENRNG and stores it in the destination register given as an argument to the instruction. The carry flag (CF) must be checked to determine whether or not a random seed was available at the time the instruction was executed.

Table 6-1. RDSEED instruction reference and operand encoding

Opcode/ Instruction	Op/ En	64/32 bit Mode Support	CPUID Feature Flag	Description
0F C7 /7 RDSEED r16	A	V/V	RDSEED	Read a 16-bit random number and store in the destination register.
0F C7 /7 RDSEED r32	A	V/V	RDSEED	Read a 32-bit random number and store in the destination register.
REX.W + 0F C7 /7 RDSEED r64	A	V/I	RDSEED	Read a 64-bit random number and store in the destination register.

Op/En	Operand 1	Operand 2	Operand 3	Operand 4
A	ModRM:r/m (w)	NA	NA	NA

7 Minimum Entropy Rate

Figure 7-1 illustrates the required minimum entropy rate and the actual observed minimum entropy rate of the noise source and the chain of conditioners consisting of a non-vetted XOR Feedback Decorrelator-Decimator and a vetted AES-CBC-MAC. The XOR Feedback Decorrelator-Decimator is a digitizer per design, but for the sake of 90B compliance, it is treated as a non-vetted conditioner.

The assessed entropy from the noise source, as per section 3.1.4.2, "Validation Testing", of the SP800-90B, is minimum (Hr, Hc, HI) = 0.6 bits of entropy per bit of data.

The CBC-MAC conditioning component is a vetted conditioning component, and Hout is assessed according to section 3.1.5.1.2, "Entropy Assessment using Vetted Conditioning Components" of the SP 800-90B.

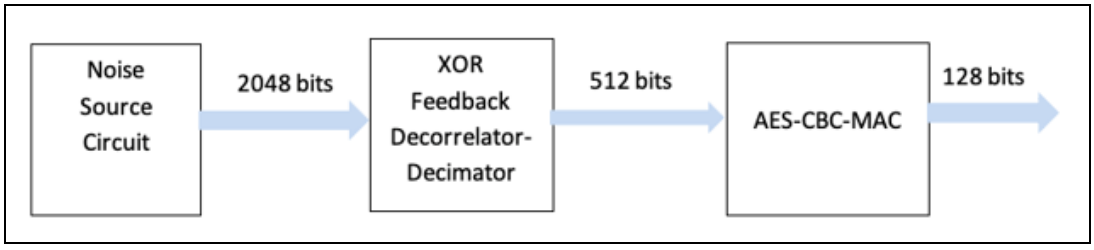
Based on IG D.K resolution, entropy source achieves full entropy meeting additional considerations:

h_{in} must be greater than or equal to n_{out} + 64 = 128 + 64 = 192

n_{out} shall be less than or equal to the security strength of the conditioning component.

The needed entropy rate must be 192/512 = 0.375.

Figure 7-1. Entropy Levels on Noise Source and Conditioning Chain



The output of the AES-CBC-MAC conditioner has the full entropy, given that the input to the AES-CBC-MAC has an entropy rate that is greater than 0.375. The ≥0.375 entropy rate requirement from the digitized noise source is based on the SP 800-90B section 3.1.5.1.1 equation for the minimum input entropy requirement of the vetted AES-CBC-MAC conditioner.

The 0.6 entropy rate on the pre-digitizer data is a design target across all variants of silicon. The actual observed entropy rate of the noise source prior to digitization on the Intel CPU is greater than 70%. The digitization stage does not reduce the entropy rate, and it is observed that the entropy rate of the post-digitizer data feeding into the AES-CBC-MAC conditioner is greater than 0.7. The difference between the 0.375 and 0.7 entropy rates is the engineering margin in the design of the entropy source.

Based on the Entropy Analysis Report, the conditioner is able to produce 256 bits of full entropy.

8 Health Tests

The DRNG includes:

- Continuous Health Tests (CHT).
- Startup Noise Source Health Tests.
- Startup Logic Integrity BIST.

The CHT is composed of a short-term test yielding a pass/fail over individual 256-bit blocks of noise source data, and a long-term evaluation of the pass/fail history of the past 256 blocks (totaling 65,536 bits) to infer an entropy source failure. The failure condition is invoked when the pass/fail rate drops below 50%.

The logic integrity test is one of the startup tests. This performs a test of the digital logic by running deterministic random sequences through all the logic and testing the resulting output against the expected result.

The Startup Noise Source Health Test involves running the CHT for a probationary period of 65,536 bits from the noise source. When the test is complete, assuming the test passes, the RNG enters the operational state. This happens during CPU startup and is completed before the first CPU instructions can be executed. Thus, `RdSeed` is available from the initial instruction execution time.

The startup test is invoked at power-on or upon exiting the reset state. Thus, the startup tests can be invoked by power cycling the CPU or resetting the CPU.

A failure of any of the startup tests (logic integrity or noise source health test) will be reflected as a BIST failure in the internal status register, which leads to an MCHECK failure of the CPU.

Following the startup tests, the CHT continues to run. Should a failure condition of the CHT test be encountered after the startup tests have completed, this may be either the result of a soft error or a hard error. The test will continue to run, and should the entropy quality return, it will exit the failure state. In the failure state, no more random numbers are issued, and the failure state is reflected in the internal state registers, which is visible at the instruction interface by the repeated failure to deliver random numbers, through the returned carry flag of the `RdRand` and `RdSeed` instructions being 0.

A sequence of 1000 back-to-back 0 carry flags on `RdSeed` can be inferred to be an error. Should some condition exist (for example, an out-of-specification operating condition), cessation of that condition will lead to resumption of the supply of random numbers.

A less computationally expensive test of failure is the response from the `RdRand` instruction (not a noise source output, but an RBG2 output), which due to its higher output rate has a 0% chance of underflows arising from high consumption demand by other threads, and so 10 back-to-back failures are specified as a runtime error.

9 Maintenance

There are no maintenance action requirements.

10 User Verification

Intel performed raw noise testing using the non-IID lower bound entropy tests of the SP800-90B Entropy Assessment software tool, showing the entropy from the noise source to exceed the minimum input threshold of 0.375 of the vetted conditioning components.

Intel performed restart testing using the SP800-90B Entropy Assessment software tool, showing the startup minimum entropy of *Hr* and *Hc* to be greater than 50% of *HI*.

Before using the RDRAND or RDSEED instructions, an application or library should first determine whether the underlying platform supports the instruction and hence includes the underlying DRNG feature. This can be done using the CPUID instruction. In general, CPUID is used to return processor identification and feature information stored in the EAX, EBX, ECX, and EDX registers. For detailed information on CPUID, refer to References (7) and (8).

To be specific, support for RDRAND can be determined by examining bit 30 of the ECX register returned by CPUID, and support for RDSEED can be determined by examining bit 18 of the EBX register. As shown in Table 1 (below) and 2-23 in Reference (7), a value of 1 indicates processor support for the instruction while a value of 0 indicates no processor support.

Table 10-1. Feature information returned in the ECX register

Leaf	Register	Bit	Mnemonic	Description
1	ECX	30	RDRAND	A value of 1 indicates that processor supports the RDRAND instruction.
7	EBX	18	RDSEED	A value of 1 indicates that processor supports the RDSEED instruction.

Appendix A Processors

Processor Name	Processor Name
Intel® Xeon® 6981E Processor	Intel® Xeon® 6710E Processor
Intel® Xeon® 6738E Processor	Intel® Xeon® 6766E Processor
Intel® Xeon® 6740E Processor	Intel® Xeon® 6731E Processor
Intel® Xeon® 6756E Processor	Intel® Xeon® 6746E Processor
Intel® Xeon® 6780E Processor	-