Public Use Document of the Intel Digital Random Number Generator Entropy Source

Intel Entropy Source

December 2024

Revision 2.0

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel is a sponsor and member of the Benchmark XPRT Development Community and was the major developer of the XPRT family of benchmarks. Principled Technologies is the publisher of the XPRT family of benchmarks. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases.

Intel, the Intel logo, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others

Copyright © 2024, Intel Corporation. All Rights Reserved.

Contents

1 **Description** 5

2 **Security Boundary** 8

3 **Operating Conditions** 10

4 **Configuration Settings** 11

5 **Physical Security Mechanisms** 12

6 **Conceptual Interfaces** 13

7 **Min Entropy Rate** 14

8 **Health Tests** 15

9 **Maintenance** 16

10 **Required Testing** 17

Figures

Figure 1: Intel 800 Series Family Chipset Family Platform Controller Hub (PCH) 6

Figure 2: Meteor Lake and Arrow Lake Platforms 7

Figure 3: DRBG Security Boundary 9

Tables

Table 1-1: Entropy Sources 5

Table 3-1: Operating Conditions 10

Revision History

Revision Number	Description	Date
2.0	• Draft to include MTL and ARL	December 2024
1.0	• Public Release	June 2024
0.5	• Chapter 10, Required Testing updated	May 2024
0.3.5	• Initial release	March 2024

1 Description

The Intel Digital Random Number Generator Entropy Source (referred to as the DRNG in this document) is a physical (P) entropy source.

The circuit component is RNG_ES_GRR_4.

The synthesizable RTL component is IP-DRNG 4.

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

The entropic data from the entropy source is not claimed to be independent or identically distributed (i.e., non-IID).

This Public Use Document (PUD) is applicable to the following processors or sub-chip cryptographic modules:

Table 1-1: Entropy Sources

Product Name	Platform / IC Name	ESV Operational Environment
Intel® Atom® P6000 Processor	Intel® Atom® P6000 Processor	CPU DRNG Intel® Atom® P6000 Processor
	Intel® Atom® P6000 Processor	FC[1] DRNG Intel® Atom® P6000 Processor
Intel® Converged Security and Manageability Engine (CSME)	Intel Core Ultra Series 1 H (Meteor Lake H)	FC DRNG MTL SoC-M Step C0 - Intel Core Ultra Series 1 H (Meteor Lake H)
	Intel Core Ultra Series 1 U (Meteor Lake U)	FC DRNG MTL SoC-M Step C0 - Intel Core Ultra Series 1 U (Meteor Lake U)
	Intel Core Ultra Series 1 U type 4 (Meteor Lake U type 4)	FC DRNG MTL SoC-M Step C0 - Intel Core Ultra Series 1 U type 4 (Meteor Lake U type 4)
	Intel 800 Series Family Chipset Family Platform Control Hub (Meteor Lake S Platform Control Hub)	CSME DRNG Intel 800 Series Chipset Step B0 (Meteor Lake S Platform Control Hub)
	Intel Core Ultra Series 2 U (Arrow Lake U)	CSME DRNG ARL SoC M Step A0 - Intel Core Ultra Series 2 U (Arrow Lake U)
	Intel Core Ultra Series 2 H (Arrow Lake H)	CSME DRNG ARL SoC M Step A0 - Intel Core Ultra Series 2 H (Arrow Lake H)

[1] FC is an abbreviation for Fuse Controller

The Atom P6000 CPU DRNG is available for general purpose compute platform applications. All other entropy sources are not general purpose and described below. The Atom P6000 FC and CPU DRNG entropy source is accessible by means of the RDSEED and RDRAND instructions.

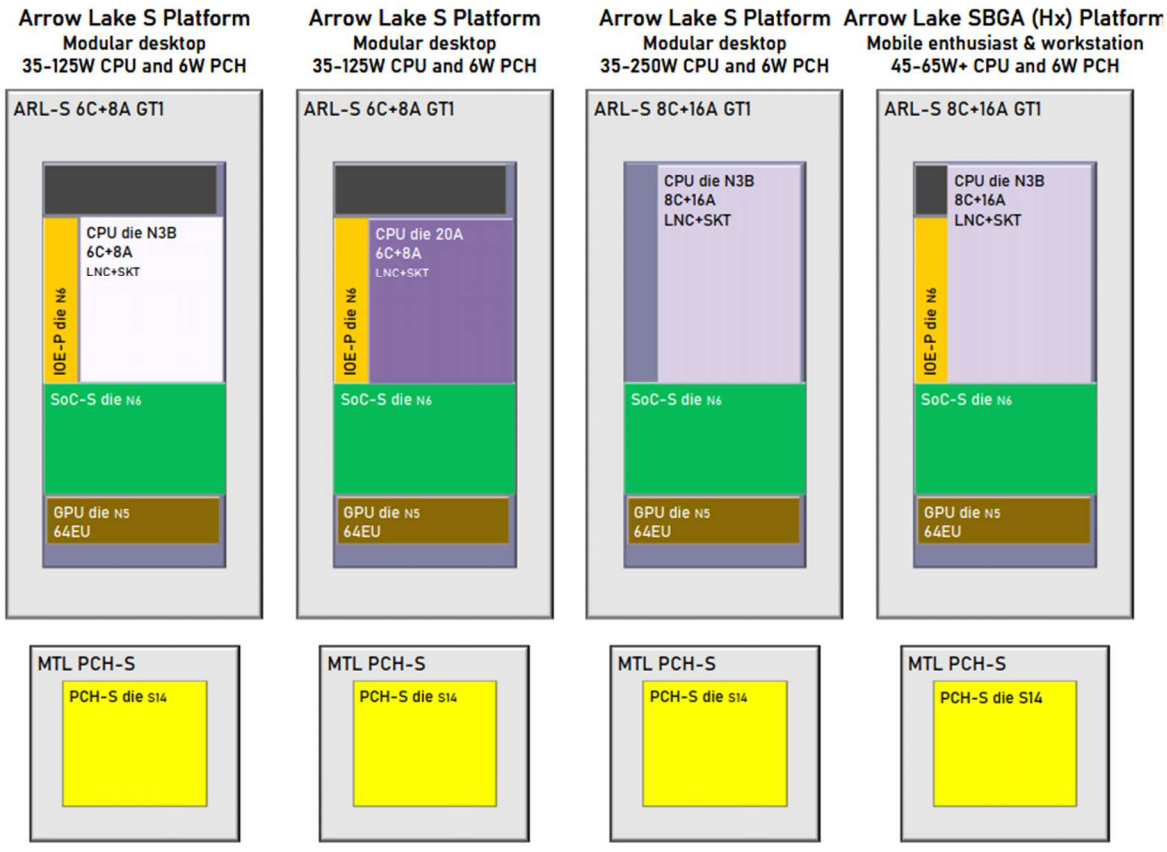
The following explanation is given to enable all parties in the certification community to easily verify the location of the CSME Cryptographic module.

The Converged Security and Manageability Engine (CSME) is an internal cryptographic module with an embedded DRNG4 Entropy Source within the module boundary. The CSME forms a deeply embedded but integral part of Intel® Meteor Lake (MTL) and Intel® Arrow Lake (ARL) platform technology.

For workstations the PCH is instantiated as the standalone Intel 800 Series IC used on the motherboard.

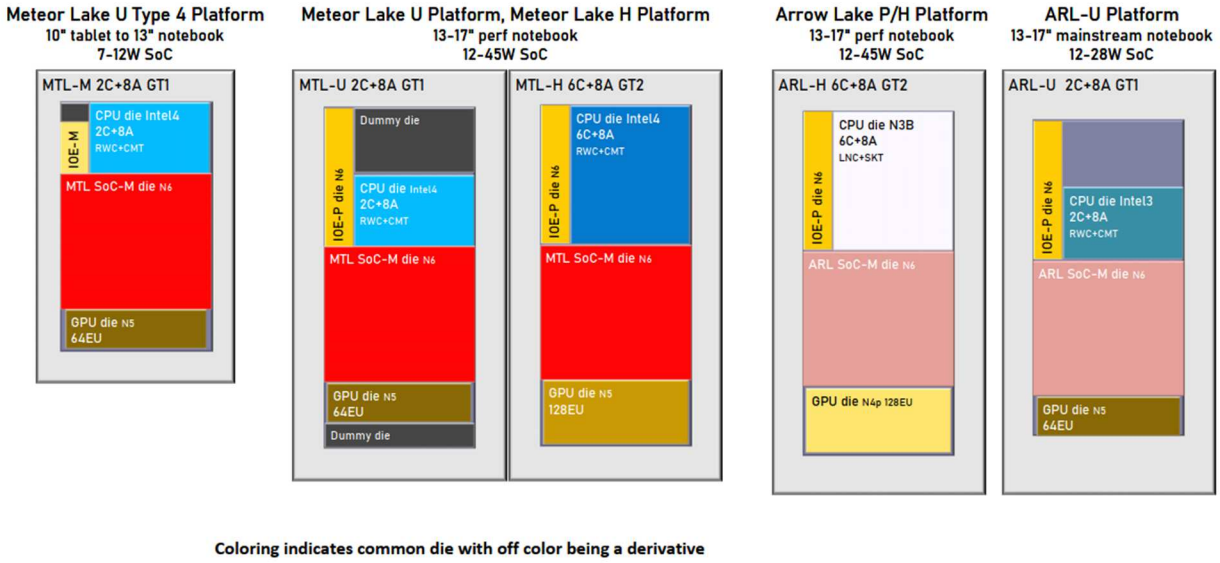
The FC DRNG MTL SoC and CSME DRNG ARL SoC entropy source are accessible by means of the GETDATA (equivalent to RDSEED) and EGETDATA (equivalent to DRAND) instructions.

Figure 1: Intel 800 Series Family Chipset Family Platform Controller Hub (PCH)



However, for mobile Meteor Lake and Arrow Lake platforms, the CSME is instantiated along with an IO Connectivity subset of what is in Intel's Platform Controller Hub (PCH) that provides 'IO Support' in the relevant SoC die as set out below.

Figure 2: Meteor Lake and Arrow Lake Platforms



Coloring indicates common die with off color being a derivative

Please note that the same MTL SOC-M Step C0 die has been used in each of the MTL platforms the DRNG Entropy Source has been tested within each platform assembly.

2 Security Boundary

The Digital Random Number Generator (DRNG) security boundary surrounds the set of components referred to as the DRNG core, that includes the noise source, digitizer, Continuous Health Tests (CHT), Advanced Encryption Standard (AES)- Cipher Block Chaining - Message Authentication Code (CBC-MAC) Vetted Conditioning Component, Deterministic Random Bit Generator (DRBG) (these are certified under CAVP cert #A5254), 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 entropy source's security boundary is a sub-boundary within the DRNG. Components outside the security boundary but within the DRNG are to attach to the local bus, clock and power systems on the chip.

The DRNG's security boundary is not a Federal Information Processing Standard (FIPS) 140-3 security boundary. It is designed to be usable within a FIPS 140-3 cryptographic module's security boundary through compliance to SP800-90Arev1, B and draft C along with relevant requirements in ISO/IEC 19790-2012 and FIPS 140-3.

The bold outline in Figure 3 shows the security boundary of the entropy source and other RNG components.

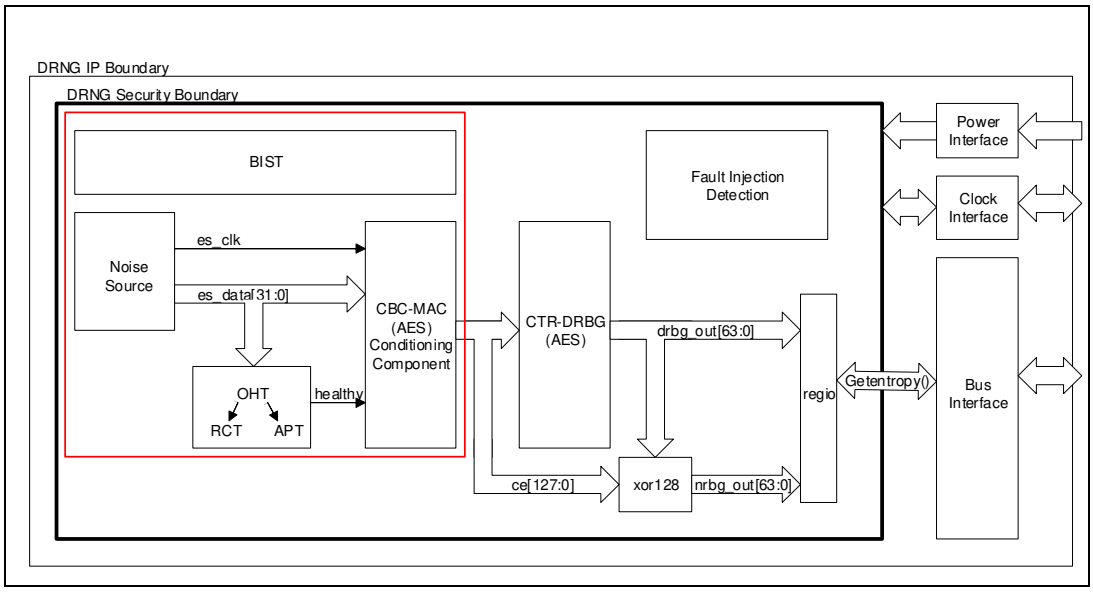
The NRBG output path to the region block is the output of the full entropy source, as an RBG3 construction.

The DRBG output path to the region block is the output of the DRBG, as an RBG2 construction, and is used as part of the RBG3 XOR construction below.

The NRBG construction is the XOR (RBG3) construction NRBG from Draft SP800-90C. This XORs the output of the conditioner with an output from the DRBG to form the NRBG output. The strength of the XOR is entirely dependent on the full entropy of the conditioner output. The DRBG's input into the XOR, as well as the XOR operation itself are not considered part of the conditioning chain.

The width of these operations is 128 bits, driven by the output size of AES. The registers' sizes are 64 bits and is transformed to 128-bits within the logic using two 64-bit entries from the 64-bit wide FIFO to match the register width.

Figure 3: 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. The silicon manufacturing process, Integrated Circuit (IC) design and DRNG design is identical for each of these devices.

3 Operating Conditions

The entropy source is guaranteed to operate within the designated operating envelope in the data sheet of the chip. Please note that the table below covers all the operating environments defined in Table 1-1: Entropy Sources above.

Table 3-1: Operating Conditions

Product	Parameter	Minimum	Maximum
Intel Atom P6000 Processor	Temperature	-40°C	110°C
Intel® Converged Security and Manageability Engine (CSME)	Temperature	-10°C	110°C

4 Configuration Settings

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

5 Physical Security Mechanisms

The DRNG contains a security boundary described in Section 2. In the operating mode of the DRNG, 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 internal state of the DRNG 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 DRNG resets itself and re-runs Built-In Self-Test (BIST). It is not possible to distinguish between a failure due to attack or environmental bounds violation or a rare false positive error. The re-running of BIST will lead to the DRNG failing if the BIST fails. In the case of an intermittent error, the DRNG 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 Intel Atom P6000 FC and CPU DRNG, `GetEntropy(n)` is implemented by the `RdSeed` instruction, where *n* can be one 16, 32, or 64 bits, depending on the size of the target register.

For the FC DRNG MTL SoC and CSME DRNG ARL SoC entropy source, GetEntropy(n) is implemented by means of the GETDATA (equivalent to RDSEED) and EGETDATA (equivalent to DRAND) instructions.

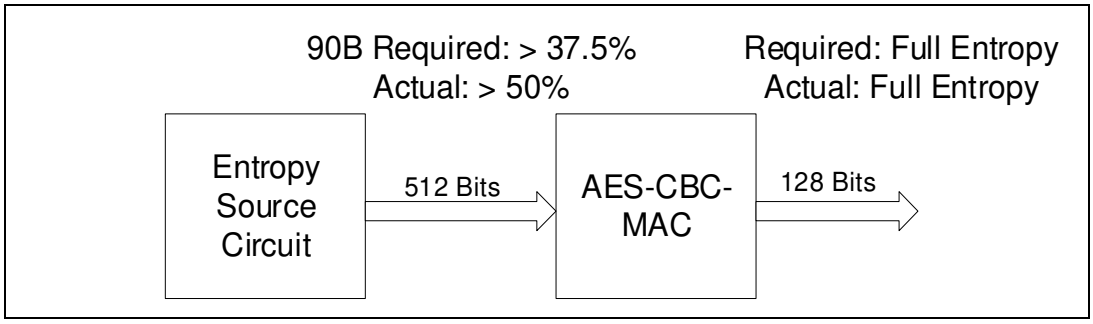
For the FC DRNG MTL SoC the consuming CSME cryptographic module must ensure the DRNG is configured to bypass the internal XOR.

Within the DRNG's security boundary, `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). This internal interface is not accessible to the consumer of the DRNG.

7 Min Entropy Rate

Figure 7-1 illustrates the required min-entropy rate and the actual observed min-entropy rate of the noise source and a vetted AES-CBC-MAC.

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



The output of the AES-CBC-MAC conditioner has full-entropy because the input to the AES-CBC-MAC has an entropy rate that is greater than 0.375 bits per bit. The requirement of ≥0.375 bits per bit is based on the FIPS140-3 IG D.K.

The 0.5 entropy rate on the pre-digitizer data is a design target across all variants of silicon. The actual observed entropy rate on 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.

8 Health Tests

The DRNG includes:

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

The vendor defined Continuous Health Test 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 block (totaling 65536 bits) to infer an entropy source failure. The failure condition is invoked when the pass 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 CHTs for a probationary period of 65536 bits from the noise source. When the test is complete, assuming the test passes, the RNG enters the operational state. This happens during startup and completes before the first instructions can execute. All the noise data used within the Startup Noise Source Health Test is discarded. The startup test is invoked at power-on or exiting the reset state. So, the startup tests can be invoked by power cycling or resetting.

A failure in 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.

Following the startup tests, the CHTs continue to run. There are two failure modes for this entropy source, intermittent and permanent. The intermittent failure mode will raise a flag in the BIST register set, but the entropy source will continue to operate. Repeated intermittent failures will result in a permanent failure, requiring a reset of the entropy source for further operation.

9 Maintenance

There are no maintenance action requirements.

10 Required Testing

No user or module integrator testing of the entropy source is required or available. The built in noise source continuous health tests and boot time hardware integrity tests verify the full functioning of the DRNG.