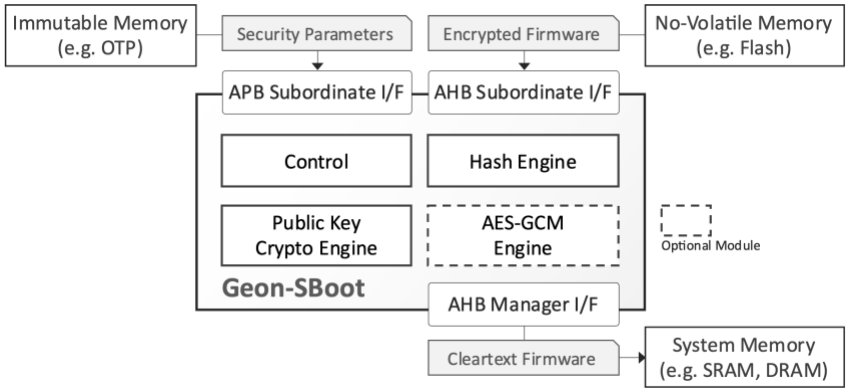
Geon-SBoot

Geon™ Secure Boot Hardware Engine

Geon-SBoot is an area-efficient, processor-agnostic hardware engine that protects SoC designs from booting with malicious or otherwise insecure code.



The security platform employs public-key cryptography (which stores no secret on-chip) to ensure that only unmodified firmware from a trusted source is used by the system. It also enables secure firmware updates over-the-air (OTA) and can prevent booting from revoked firmware versions. Optionally, Geon-SBoot can use symmetric encryption to protect the confidentiality of the firmware and prevent other devices from running firmware clones.

Designed for straightforward use in nearly any SoC, Geon-SBoot works with all modern architectures including RISC-V and ARM. It requires no software assistance from the host CPU, is independent of the memory types used, and uses standard interfaces. It further gives designers great flexibility in the boot control flow.

The isolated Geon-SBoot subsystem interfaces to the host system via three AMBA® ports: a subordinate AHB port for receiving the encrypted firmware, an AHB manager port for writing the authenticated, decrypted firmware to the system's memory, and an APB subordinate port for receiving the security parameters. The security parameters (i.e., a hash of the public key and the symmetric key if used) are typically stored in immutable memory, constituting the root of trust. Geon-SBoot reports boot success or failure on its status register and via dedicated interrupt lines. It can optionally make its crypto accelerators available to the host system post-boot.

The Geon-SBoot core is production-proven and adheres to the industry's best coding and verification practices to ensure trouble-free implementation in ASIC or FPGA technologies.

Performance and Size

Hardware acceleration of the computationally complex cryptographic algorithms makes Geon-SBoot exceptionally fast, imposing a competitively-small impact on boot time. The boot time overhead depends on the configuration (e.g., length of the public key) and the implementation decisions (e.g., clock rates) and it can be sub-millisecond for a basic configuration using RSA 2048 on a modern technology node. Despite this speed, the Geon-SBoot core does not require excessive silicon resources thanks to its efficient design. Its size starts from 50k gates and depending on its configuration it requires just 8k to 11k bytes of internal memory.

Integration and Customization

The Geon-SBoot engine can be pre-integrated with RISC-V processors and other IP available from CAST and customized to meet specific project requirements. Please contact CAST Sales to learn more.

FEATURES

Protection Layers

- Ensures integrity and authenticity of boot image
- Prevents any firmware downgrade (anti-rollback)
- Optionally protects confidentiality and prevents firmware cloning
- Complete software and hardware isolation from the host SoC

Public-Key Authentication Benefits

- No secret on die for resistance to physical and side-channel attacks and easy deployment
- Over The Air updates
- Immune to break-one break-all scenarios

Fast & Compact

- Minimal boot time impact: typically from sub-ms to a few ms
- From 50k gates and 8k to 11k bytes of memory

Cryptographic Algorithms

- Asymmetric authentication
  - RSA signature verification with key lengths 2,048, 3,072 and 4,096 bits (default)
  - ECC signature verification with NIST-validated (i.e. P-224, P-256, P-384, & P-521) or custom curves (option)
- SHA3-256/384/512 Hashing or SHA-224 (option)
- Symmetric authenticated decryption with AES-GCM with key length of 128 or 256 bits (option)

Easy to Use and Integrate

- AMBA AHB and APB interfaces
- Autonomous & isolated operation requires no software assistance.
- Protected firmware can be broken in fragments, each stored at different memory regions. Multi-stage boot support
- Crypto acceleration engines can be made available to the system after boot

Reusable & Portable

- Processor-Agnostic: works with any host processor(s) or SoC
- Process-Independent RTL design

Deliverables

- Verilog RTL source code or targeted FPGA netlist
- Comprehensive Documentation
- Software tool for signing and encrypting boot images