Intel FPGA Secure Device Manager

Karen Horovitz, Ryan Kenny
Intel Corporation, Programmable Solutions Group
101 Innovation Drive, San Jose CA 95134
{karen.horovitz, ryan.kenny}-@intel.com

***Abstract*— In previous generations of FPGA devices Intel Programmable Solutions Group (formerly Altera) employed security features based on general bitstream protection and anti-tamper requirements. As military applications and the general technology landscape evolve, more data is being processed resulting in a larger number of attack points. To resolve a response to these threat vectors, a central security controller is necessary. The secure device manager, or SDM, is a hardware processor which provides an evolvable core of device security and allows an end user greater flexibility in threat response as well as application design and control. Several different security functions can be executed through the SDM which primarily include zeroization, sectorization, PUF key encryption, and anti-tamper control. Furthermore, the SDM contains field upgradable firmware that will follow the Intel threat mitigation strategy. This paper will discuss the features of the SDM, future product roadmap in which upgraded SDM security will be integrated, as well as a lifecycle of the mitigation strategy process, and military and government use cases.**

Keywords—FPGA; SoC; security; encryption, Stratix; SDM

I. Introduction / Background

FPGA device security has been historically based on protecting the confidentiality, and more recently, integrity, of the FPGA bitstream where all user intellectual property resides. Using encryption and authentication had been satisfactory for mitigating threats. Toward this end, on-chip security design in past generations of Intel PSG (formerly Altera) devices were focused solely on bitstream protection. In Table I, basic security features are shown in past devices, from Cyclone III LS to Stratix 10/SoC.

Today an evolved security landscape lends itself to different needs. This evolution is shown with encryption, which originally was a requirement driven by military customers. Several commercial end customers have become interested in bitstream protection in recent times, specifically within industrial and automotive markets. The current landscape lends itself to connectivity in multiple forms - through enterprise edge networks and data center to the cloud or within secure communications through satellite links. Infotainment systems in autonomous vehicles connect to consumer devices and need data link protection. Emerging military technologies such as government analytics algorithms need focus on cybersecurity, which includes anti-tamper and digital data protection. As device generations have matured, Intel PSG has added additional capabilities that have surfaced as necessary to support these market needs as well as address published attacks. The basic added requirements are shown in *Table 1* under the Generation 10 devices.

Table I. Intel FPGA Security Features [1]

Device Security Features	**Device**				
	Cyclone III LS	**Stratix V Arria V/SoC Cyclone V/SoC**	**Max 10**	**Arria 10/SoC**	**Stratix 10/SoC**
Bitstream Encryption and Authentication (AES-256/SHA-256)	Encrypt	Encrypt	Encrypt	Encrypt/ Auth	Encrypt/ Auth
Boot Code Auth (SoC)	N/A	No	N/A	ECDSA 256	ECDSA 256/384
Vol and Non-Vol Key Storage	Volatile	Both	Non-Volatile	Both	Both + PUF
Side Channel Attack Protection	No	No	No	Yes	Yes
Readback, JTAG, Test Mode Disable	No	No	No	No	Yes

II. Stratix 10 and Beyond: Security Overview

With the addition of Stratix 10/SoC and other devices, Intel PSG has realized that the market needed a failsafe, strongly authenticated but programmable security scheme, with modern encryption blocks and hardware-based identity. Intel has recognized these challenges and requirements across users of FPGA security features, and responded with the design of the security architecture of Intel® Stratix® 10 FPGAs and SoCs (formerly Altera® Stratix 10 FPGAs and SoCs).

Stratix 10 SoC and FPGA devices include bitstream encryption and authentication, volatile and non-volatile key storage with additional protection available using a Physically Uncloneable Function (PUF), JTAG and test mode disable and/or authentication-only mode, and tamper detection sensors and monitors (voltage and temperature). Side channel attacks are addressed through a series of microcontroller-based countermeasures which in turn can themselves be modified and upgraded in response to vulnerability analysis and threat mitigation reporting. ECDSA will naturally resist DPA attacks due to it only protecting a public key on the HPS (Hard Processor System) side.

Attacks that substitute or alter the contents of flash storage that include ARM boot code or FPGA bitstreams can be prevented by strong authentication into the HPS system or through security fuses.

One major architectural update to the Stratix 10 (and future Intel-based FPGA products) security feature list is the SDM, or secure device manager, of which a high-level view is shown in Figure 3 below.

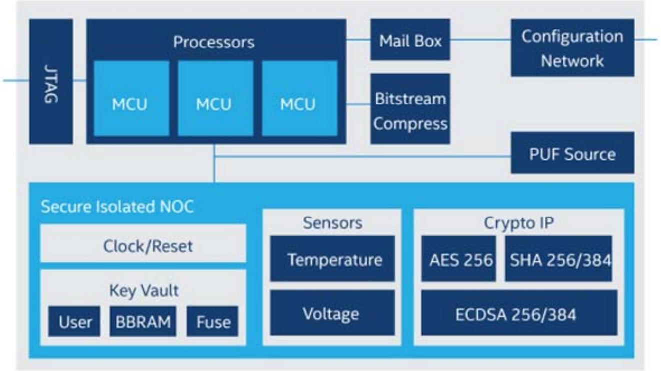

Fig. 1. Secure Device Manager Functional Blocks

III. SDM Architecture [2]

The SDM processor acts as the configuration manager and security enclave of the FPGA device and can conduct reconfiguration and security functions through JTAG commands and user logic interfaces to the SDM. External data, in the form of first firmware code for the SDM itself and then the user configuration data, enters the SDM to be authenticated with one or more digital signatures. The SDM operation does not affect timing closure of the user design as it functions out of band from the user logic. Once configured, the SDM conducts functions such as those mentioned in Table 2; specifically, sectorization, PUF key protection, key management, hard encrypt/authenticate engines, and zeroization. Additionally, environmental sensors monitor voltage and temperature; these sensors and access to the sensors from user logic are controlled by SDM. Device maintenance functions such as secure remote update (authenticated) and secure return material authorization (RMA) without revealing encryption keys, secure debug of designs and ARM code, and secure key management are all additional functions enabled by the SDM.

A. Sectorization & Zeroization

One of the architectural features that enables the SDM functionalities as well as various SDM use cases is that the Intel FPGA fabric has been split into configuration sectors. After configuration data, on a sector block basis, is authenticated and decrypted, these data blocks are distributed to respective sectors in parallel, shown in Figure 4. This allows for additional flexibility which will be discussed in the use cases section.

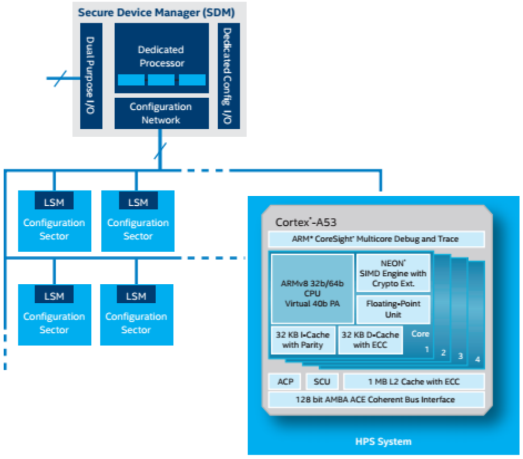

Fig. 2. Configuration Divided into Subsystems and FPGA Fabric is Further Divided into Sectors

FPGA configuration sectors are of fixed size and overlay normal rows and columns of routing logic so as to not impact timing and place and route. Within each sector lies a Local Sector Manager (LSM), which is an additional microprocessor that parses configuration block data and configures logic elements. The LSM continues to monitor for single event upsets at the sector level, can process scripted responses to SEUs, and can perform hashing or integrity checks as needed.

Configuration management of the FPGA by sectors creates several additional security capabilities to this FPGA architecture. In addition to seamless partial reconfiguration capabilities when users design to these sector boundaries (though they are not required to adhere to sector boundaries), the sectors may also be rewritten, reauthenticated, and zeroized through SDM commands or as a default response to a specified threat.

B. PUF Key Protection

Intel Stratix 10 FPGAs enable user access to a PUF as part of the device configuration process for key protection and key material generation, or for device identification purposes. This SRAM based PUF generates device-unique, unclonable keys that designers can use for device authentication and key wrapping. Intel selected this PUF technology and algorithm from partner IntrinsicID, based on early characterization data of SRAM cells generated on Intel's 14 nm process technology.

The IntrinsicID PUF algorithm runs as instruction code in the SDM, and is only incorporated in SDM configuration for IntrinsicID licensees. The PUF function relies on dedicated hardware sources of entropy, but also has a software updateable algorithmic component to address changes and fixes resulting from longer lifetime characterization activities. This approach also allows for nearly unlimited PUF reregistration by users.

The SDM-executed firmware PUF has substantial advantages over hard PUF circuit approaches for FPGAs and other microelectronics that rely on leading-edge manufacturing technologies. It provides a firmware-based methodology for algorithmic tuning and optimization as more characterization data becomes available.

C. Whole Bitstream Authentication

Authenticating a data block as large as modern FPGA bitstreams has several challenges, including data buffering and potential impacts on configuration time.

Running one continuous hash or authentication function on an FPGA bitstream requires a continuous operation on streaming configuration data. This constrains the user to configure the FPGA in the same order on every boot-up, as well as creating temporal restrictions on the order of authentication and decryption. In the Stratix 10 security architecture, individual bitstream data blocks are hashed and authenticated (similar to an HMAC process), but then the data hashes themselves are authenticated and used to authenticate the next block of data and its hash digest. This process provides enough parallel processing to maintain low configuration times for a fully authenticated bitstream, but also enables portions of the bitstream chain to be reloaded for partial reconfiguration operations with the full authentication capabilities of an initial configuration.

In the diagram below (*Figure 3*), both authentication and decryption mechanisms are shown using a user public key (in FPGA) and user AES root key (in FPGA). A bitstream header file is received and authenticated, containing hash digests for the first FPGA bitstream sector image(s), which are loaded in fixed or pre-determined orders, and include hash digest files for their own sector as well as the next sector in configuration order. The original bitstream header also potentially includes a user sub-root key file of sector encryption keys protected by the user AES root key. Additional details of this authentication process will be included in final documentation of the Stratix 10 Security User's Guide.

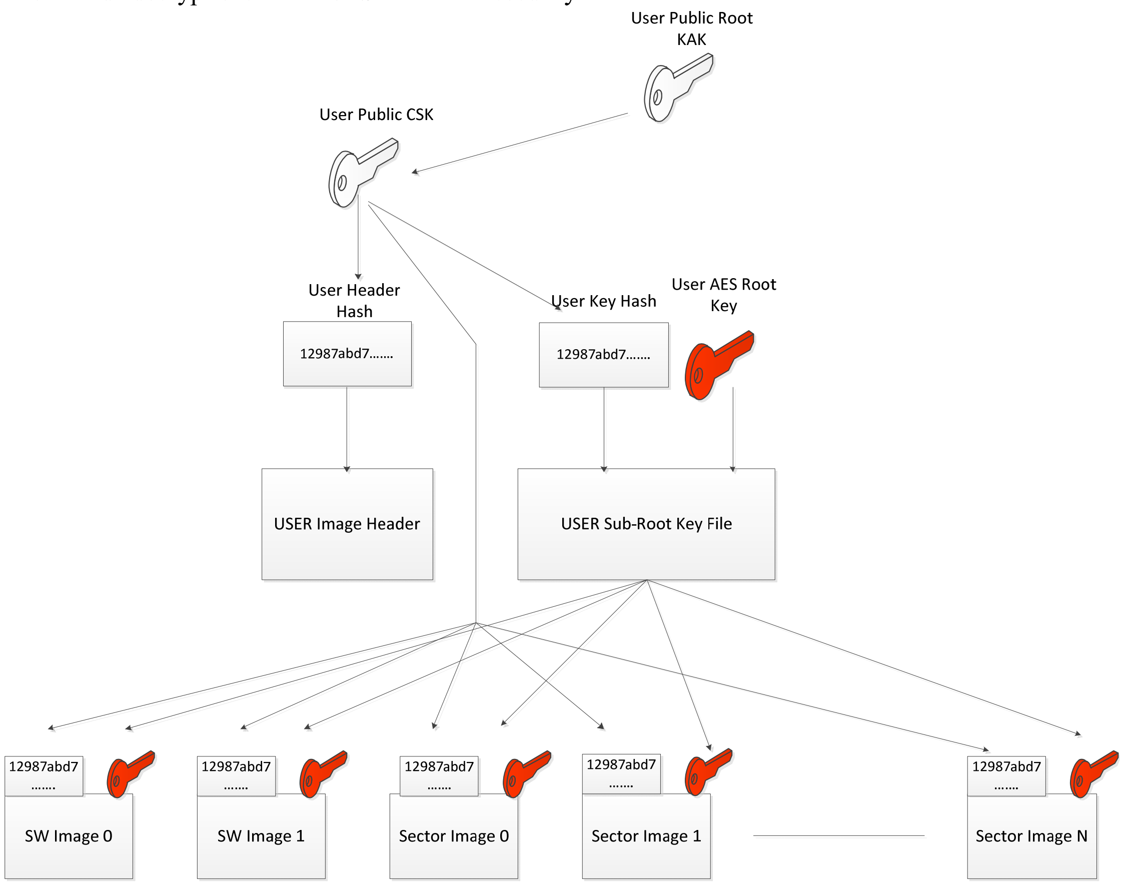

Fig. 3. Key Hash Derivation Flow for Full Bitstream Authentication

IV. Mitigation Strategy Process

The Intel Secure Design Lifecycle (SDL) is the process that integrates security and privacy activities into Intel's Product Life Cycle (PLC). [4] The stages of the SDL process include exploration, planning, development, and production: these are shown in *Figure 4*. Any relevant security proposals or potential threat mitigations must comply with this lifecycle – specifically the requirements of the post-silicon, product security incidence response process.

Toward this end, any SDM modifications or firmware updates will comply with SDL process, allowing for end customer trust in threat mitigation and response.

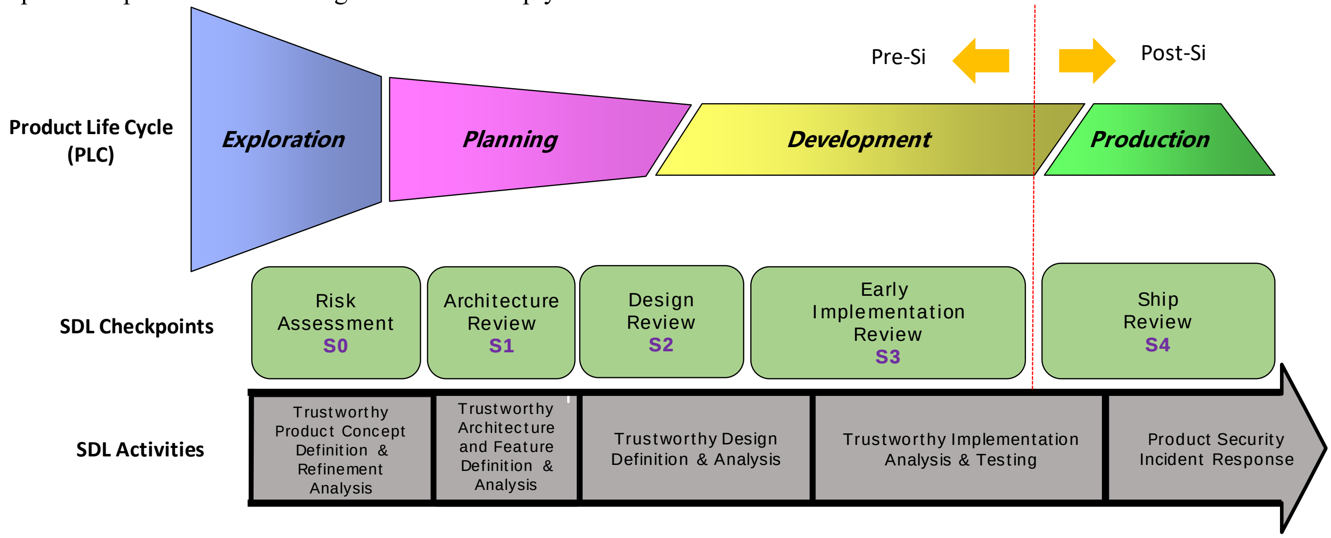
Pre-Si
Post-Si
Product Life Cycle (PLC)
Exploration
Planning
Development
Production
SDL Checkpoints
Risk Assessment S0
Architecture Review S1
Design Review S2
Early Implementation Review S3
Ship Review S4
SDL Activities
Trustworthy Product Concept Definition & Refinement Analysis
Trustworthy Architecture and Feature Definition & Analysis
Trustworthy Design Definition & Analysis
Trustworthy Implementation Analysis & Testing
Product Security Incident Response

Fig. 4. SDL Process at Intel

A. SDL Process For SDM Field Upgradability

There are two separate opportunities for updates to the firmware code that drives the SDM in Stratix 10 FPGAs. The first is for 'patching' new security capabilities into Stratix 10 FPGAs according to evolving standards or customer requirements. Examples of this can include new requirements or standards for device zeroization, novel techniques for error generating PUF values from raw SRAM entropy, and key rotation algorithms or methods. The other is to directly address reported vulnerabilities or concerns with the device configuration security according to the SDL process. System vulnerabilities, discovered within Intel, by third parties, or by Government sources, are collected and examined by the Intel PSG Security Team and vetted by Intel's Security Center of Excellence (SeCOE). Mitigation strategies will be developed, tested, and scheduled for release within the Quartus II ACDS design software. Customers elect whether to receive these updates by recompiling their designs in the new version of Quartus, which in turn will pack the user design database with the new version of SDM firmware.

B. SDL Process for Third Party IP

Only one component of the SDM firmware is sourced from a third party and that is the (licensed) PUF conditioning code. This component of the firmware will be updated according to characterization and vulnerability assessments of the PUF, and integrated into upgrades in Quartus according to a separate third party IP vetting process [4].

C. Intel Support for Customer Mitigation Strategy

For customer designs that undergo vulnerability testing, any potential vulnerabilities identified can be shared with Intel and become part of Intel's PSIRT and SDL process. In addition, direct and remote bitstream upgrades will be supported in future versions of Quartus II for Stratix 10.

V. Use Cases

As mentioned previous, there are several use cases for the SDM, both for commercial and military. These draw upon features of the SDM such as secure authenticated firmware updates, sectorization, zeroization. All use cases follow SDL process for optimal threat mitigation strategy.

A. Key Management and Encryption Key Updating

One of the advanced use cases enabled by the SDM is key management and encryption key updating for Intel Stratix 10 FPGAs and SoCs. In this case, either the SDM code itself, or an external command authenticated by the SDM, introduces new encryption key material into the SDM cache memory, retires or replaces encryption key material, and generates new encryption key material. Encryption keys can be used for securing and authenticating communication with external devices, for encrypting and decrypting sector configuration data, or applying new signatures to data processed within the FPGA. Encryption key updates can be effective as long as the device is powered; however, persistent (accessible after device reset) encryption key updates must be overwritten into the device configuration flash off-chip. The encryption key fields available for device root key include battery-backed RAM, one-time programmable fuses, and the PUF function. Additionally, the SDM can store a user encryption key vault of keys during configuration (and be updated securely later).

Remote key updates are useful in situations such as side channel attacks. In order to recover the device, the SDM must be updated with a new key.

For military scenarios with radar, electronic warfare, or missiles that have been deployed into the field, potential threats may either disable or alter the desired functionality. In these use cases, the SDM key management scheme will be utilized.

B. Design Separation

Design separation, done utilizing FPGA sector boundaries, is another area that adds substantial generational security capabilities with the SDM. Design separation implies first that there is clear data and transmission line separation between FPGA design regions (which may or may not be Stratix 10 FPGA configuration sector boundaries) and in the cloud and MLS use cases, different security properties for each design region.

1) Third-Party IP and Designs

In military commercial off the shelf (COTS) boards, vendors may like to secure part of their design that is proprietary and allow end users to have flexibility in securing their input design within the remainder of the FPGA fabric. This is possible with sectorization and separately keyed partial reconfiguration regions/sectors.

In the commercial market, this is interesting to cloud and data center vendors who have customers writing to a shared space. A sector may be allocated for user design, or green bitstream, whereas blue bitstream would contain proprietary company data. This is a similar model to user and kernel read/write access in hardware.

2) Multi-Level Security

A case that may be of interest to military applications is that in which a user design is split into sectors or modules and implemented into partial regions or sectors based on certain security classifications, such as country-specific, secret, top-secret, and compartmentalized data.

In previous devices such as Arria 10, this was done using an internally developed design, Partition-Based Security[5]. The module with lower security level is associated with a non-volatile key and the modules with higher security levels are associated with different volatile keys. Access to a successive key is based on the previous key and a user designed authentication function in the initial region, such as a PUF or certificate exchange.

In Stratix 10 and future devices, using the SDM sectorization capability, sectors could be designed with different levels of security, with separate sub-root keys that are generated from an initial AES root key. These sub keys can either be user-defined or generated with key expansion algorithms.

C. Enabling Technologies like Physically Unclonable Functions

The secure levels use case will provide a host of additional services to a designer of military system such as physically unclonable functions, key provisioning and management, and SEU (single event upset) management.

It is possible to implement a soft PUF into one region in order to authenticate another region or sector. The PUF can produce the key for the subsequent region or the PUF can authenticate through the initial region to determine if the right user has authenticated the right silicon. Similarly, off chip certificates or entropy could be used in place of a PUF for this purpose.

VI. Conclusions/Future

The SDM provides novel security capabilities for FPGA devices, catering to a market that must respond to a larger number of threats than in the past. The security use cases discussed are applicable both to users who are concerned with implementing higher security levels in parts of their design as well as those partners who require design separation security. However, many other security use cases exist in which multi-key configuration is utilized.

Design separation security can also be utilized in commercial security applications. Within the data center, FPGA's may serve as accelerator platforms for multiple cloud instances, with each accelerator PR region requiring separate key protections. In automotive applications, parts of the design are deployed to different vendors or collaborators. In some cases, the end user may want to design modifications without affecting the existing design.

For future devices, there are more possibilities. Implementing a larger number of crypto and authentication accelerators that could protect one or more sectors could expand upon secure levels use cases. This implementation could also support multi-key configuration and blue-green separation use cases for the commercial market. These accelerators will follow the SDL process through security assurance by the third-party IP vendor [4].

Intel PSG is now supported by Intel Labs and the distributed research grants of Intel Research. Many of the research areas inside and outside Intel are focused on security to include hardware anti-tamper capabilities and advanced encryption and key management blocks (to include encryption to memory). Much of this research will intersect with the future technology roadmap of Intel PSG FPGAs.

VII. References

[1] *Comparing Altera SoC Device Family Features*, http://www.altera.com/literature/hb/soc-fpga/UF-01005-2014.01.15.pdf

[2] Lu, Ting, Kenny, Ryan, Atsatt, Sean. *Secure Device Manager for Intel Stratix 10 Devices Provides FPGA and SoC Security*. 2016. https://www.altera.com/en_US/pdfs/literature/wp/wp-01252-secure-device-manager-for-fpga-soc-security.pdf

[3] Application Note 556: Using the Design Security Features in Altera FPGAs

[4] Brent Sherman, et al. *Security Assurance Guidance for Third-Party IP*. J Hardw Syst Secur (2017) 1:38-55.

[5] Karen Horovitz, et al. *Protecting Partial Regions in FPGA Bitstreams.* IEEE IVSW 2017.