WHITE PAPER

FPGA

Secure Device Manager for Intel® Stratix® 10 Devices Provides FPGA and SoC Security

The Secure Device Manager for Intel Stratix 10 devices provides a failsafe, strongly authenticated, programmable security scheme for device configuration.

Authors

Ting Lu
Senior Security Architect
Intel® Corporation

Ryan Kenny
Senior Strategic Marketing Manager
Intel Corporation

Sean Atsatt
Senior Configuration Architect
Intel Corporation

Table of Contents

Introduction

Over the last ten to twenty years, all major FPGA component providers have invested in security features to protect their users' proprietary and sensitive designs. These features have existed for several generations of FPGA families, and primarily focus on the encryption and later authentication of configuration bit streams. Over time, many of these features have proven valuable while others have shown themselves to be vulnerable to published attacks and probing techniques.

Just as the explosive growth of cloud computing, software as a service, and the Internet of Things (IoT) have introduced entirely new classes of threats to the Internet (i.e., cyber security), the complexities of FPGA products and customer designs have contributed to an increase in potential malicious attacks on FPGAs and SoCs.

Despite FPGA company investment in new security capabilities and structures, the fixed and predictable nature of the device configuration process itself is an untapped area of security investment. In both SRAM and flash device configuration processes, fixed state machines manage the order of authentication, decryption, decompression, and actual device configuration. What is needed is 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.

Introducing 'configurability' to configuration

Recognizing this FPGA design security issue, Intel Arria® 10 SoCs introduce the industry-unique capability for user-selected boot order. This method allows specific applications, or configuration loads of the Intel Arria 10 SoC to select whether the FPGA design or HPS system application configures first, and whether configuration control of the second system is managed by the first. This scheme gives the SoC designer a flexible, first order degree of control over the Intel Arria 10 SoC configuration parameters.

Intel Stratix 10 FPGAs and SoCs, built on Intel's 14 nm Tri-Gate transistor technology, (1) offer the next dimension of flexibility and user-selected configuration control with the Secure Device Manager (SDM). The SDM is a microprocessor block available in all densities and variants of Intel Stratix 10 FPGAs and SoCs that provides a robust, secure, and fully authenticated configuration scheme. Additionally, it allows users to customize device configuration. Other advantages include configuration time, responses to single-event upsets, reactive zeroization of data as a security response, key management and update, and providing field upgrades. This combination of features and flexibility in the SDM for Intel Stratix 10

devices provide security to protect sensitive intellectual property (IP) and data in both FPGA and SoC devices.

Overview of the secure device manager for Intel Stratix 10 devices

Figure 1 provides a high-level summary of the SDM functional blocks. Not all functions are discussed in this white paper. Refer to the Intel Stratix 10 device technical documentation and the Intel Stratix 10 TX Advance Information Brief[2] for additional details.

The SDM is the point of entry to the FPGA for JTAG commands and interfaces, as well as for device configuration data (from flash, SD card, or through PCI Express* hard IP). The first component of configuration data that enters the SDM is the configuration data and microcode for the SDM itself, which is authenticated with one or more digital signatures (see "Configuration Process" on page 5). Once the SDM is configured and the processors are released from reset, the SDM block manages all Intel Stratix 10 FPGA or SoC security and configuration functions. This management occurs out of band from the user design, and does not affect timing closure or any other parameters of logic design.

SDM-enabled security functions

New security features have been introduced with each generation of FPGA and SoC products. Table 1 provides a top-level overview of these features. Intel Stratix 10 FPGAs continue to support these features, including bitstream encryption and authentication, volatile and non-volatile key storage, JTAG and test mode disable, and tamper detection sensors and monitors (voltage and temperature).

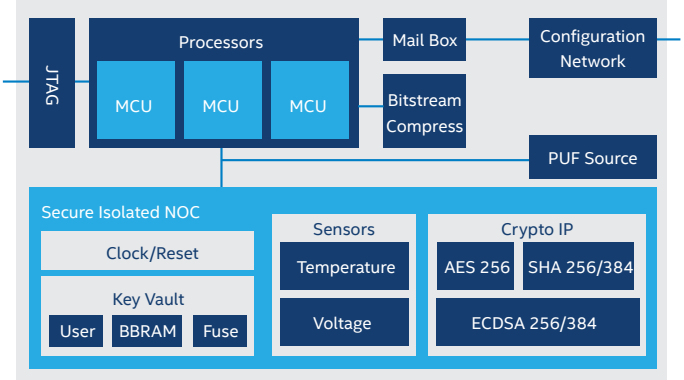

Figure 1. Secure Device Manager Functional Blocks

Device Security Features	Cyclone® III LS FPGA	Stratix V FPGA Arria V FPGA/SoC Cyclone V FPGA/SoC	Intel MAX® 10 FPGA	Intel Arria 10 FPGA/SoC	Intel Stratix 10 FPGA/SoC
Bitstream Encryption and Authentication (AES-256/SHA-256)	Encryption	Encryption	Encryption	Encryption/ Authentication	Encryption/ Authentication
Volatile and Non-Volatile Key Storage	Volatile	Both	Non-Volatile	Both	Both plus PUF
Boot Code Authentication (SoC)	N/A	No	N/A	ECDSA256	ECDSA 256/384
Side Channel Attack Protection	No	No	No	Yes	Yes
Readback, JTAG, and Test Mode Disable	No Readback, JTAG Disable	No Readback, Fuse JTAG Disable	No Readback, Fuse JTAG Disable	No Readback, Fuse JTAG Disable	Readback and JTAG Fuse Disable
Dedicated Secure Configuration Processor	No	No	No	No	Yes, see Table 2

Table 1. Overview of Security Features Offered by Intel FPGA Products

Intel Stratix 10 Device New Security Features	Feature Value		
	IP Protection	Glitch/Malware Protection	Tools for Secure System Design
Sector-Based Security	Allows Multi-Key IP Protection	SEU/Glitches Detected at Sector Level	–
SRAM PUF for Key Protection and Hardware Identity	Additional Source of Key Protection	–	Unique ID Available for Hardware Authentication
Flexible Key Management and Generation Capability	Broader Source of Key Material	Device/Key Updates After Compromise	Tools for Key Management Applications
Hard Encryption/Authentication Engines	–	Hashes and Signatures to Detect Modification	Hard IP Blocks Available to FPGA Designers
Programmable and Updateable Configuration Process	Update Configuration After Compromise	Update Configuration After Compromise	–
Command Authentication for Remote Update and Key Management	Secure Update of Device	Active Prevention of Malicious Updates	Field Upgrades
Scripted Responses to Sensor Inputs, including Tailored Zeroization	Zeroize keys when attack detected	–	Part of Larger Security and Tamper Resistance Solution

Table 2. New Security Features and Use Cases

The addition of a failsafe dedicated secure configuration processor in Intel Stratix 10 FPGAs with associated hard accelerator and physically unclonable function (PUF) service blocks, introduces whole new classes of security features as shown in Table 2. These features are described in subsequent sections.

Triple-redundant processor

The core SDM functionality is a triple-redundant microprocessor that executes commands on a voting circuit of all three cores. With this arrangement, the SDM function is inherently resistant to environmental or intentional glitches and single-event upsets. This feature of the SDM is important because this subsystem is responsible for receiving input from environmental sensors and single-event upset detection logic, and providing scripted responses to events in which device operation or integrity can be affected.

Environmental sensors

The SDM block contains sensors for on-chip temperature measurement and monitoring device voltage rails. The SDM can be configured to provide security automatically or to provide safety responses to changes in these variables as part of device configuration. Sensor monitoring and event responses occur in the SDM independent of user logic.

Encryption support hardware

The SDM contains several encryption and authentication support blocks to support the system's bitstream configuration functions:

- AES 256 Encrypt/Decrypt Accelerator Block
- SHA2 256/384 Accelerator Block
- ECDSA 256/384 Accelerator Block

The SDM uses these blocks during the configuration (and reconfiguration) process; however, these accelerator blocks are also available for user applications after device configuration with appropriate licensing through the Intel Quartus® Prime software. For example, designs could use these blocks for encryption/decryption of data traffic in user applications, as well as authenticating messages to and from the FPGA. Service requests to these blocks come through either the FPGA fabric or the ARM* Cortex*-A53 HPS system to a request mailbox in the SDM, which instructs the blocks to perform encryption, decryption, hashing, signing, or signature checking functions.

The encryption and authentication support blocks are licensed from The Athena Group, who also provide documentation and NIST sourced certifications for these cores.

Strong secure hashing algorithm (SHA) and elliptic curve (ECDSA) blocks provide layered authentication that supports multiple signatures using different keys on sensitive data blocks. In contrast, multiple signatures are not available using authentication-enabled encryption methods like AES-GCM. Elliptic curve algorithms also boast significant key length versus signature strength advantages over other asymmetric algorithms such as RSA.

Physically unclonable function for key material, protection, and identity

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. The PUF generates a device-unique, unclonable key that designers can use for device authentication and key wrapping. More details on PUF use cases for configuration and IP security will be provided in the upcoming Intel Stratix 10 device technical documentation.

The PUF technology enabled in select Intel Stratix 10 FPGAs and SoCs is based on a unique, unclonable SRAM initialization pattern, which is available to the SDM through a dedicated field of SRAM cells that are powered up only when generating a PUF value.[3] Intel selected this PUF technology and algorithm from partner IntrinsicID based on superior characterization data [4] of SRAM cells generated on Intel's 14 nm process technology.

The IntrinsicID PUF algorithm runs as an 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. Intel believes the SDM-executed PUF to be superior to 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.

Advanced key management schemes enabled by secure processor

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 later).

Advanced device maintenance functions enabled by secure processor

The SDM user and command authentication capabilities enable a class of new secure device maintenance functions for Intel Stratix 10 FPGAs and SoCs. These functions include secure remote update (authenticated), secure return material

authorization (RMA) of devices without revealing customer encryption keys, secure debug of designs and ARM code, and secure key management. These use cases will be further described in upcoming technical documentation.

Sector-based configuration

One of the primary Intel Stratix 10 device architectural features that enables these SDM use cases is the logical separation of the device into configuration sectors. Dividing FPGA configuration into logical sectors helps manage configuration times and bottlenecks when configuring very large devices. After configuration data is authenticated and decrypted using the high-performance encryption accelerator cores, configuration data blocks are distributed to the various sectors in parallel on a configuration network. This sectorization and data distribution network provides distinct flexibility advantages.

Sector layout

Figure 2 shows the division of logic element configuration by sector across a configuration network. FPGA configuration sectors are a fixed size across the Intel Stratix 10 device family, allowing for natural design boundaries for IP re-use, security, and reconfiguration. The sectors are logical for configuration purposes, but otherwise overlay the normal rows and columns of routing logic; i.e., there is no impact to Intel Quartus Prime software place and route or logical timing from logic and data paths that cross sector boundaries.

Local sector manager

Within each sector is another microprocessor called the Local Sector Manager (LSM). The LSM parses sector configuration block data and configures the logic elements for each sector. After configuration, these microprocessors monitor for single event upsets at the sector level, process scripted responses to these SEUs, and can perform hashing or integrity checks in real time (out of band of the user design) for real-time configuration integrity.

Sector-based reconfiguration

Because the device is divided into logical sectors, you can quickly reconfigure a portion of the Intel Stratix 10 FPGA design. Because you configure the FPGA by logical sector, you can use a subset of this configuration process to reconfigure a subset of the sectors. The SDM can command and execute this sector-based reconfiguration, out of band from the user design.

Zeroing design information by sector or in parallel

Zeroing encryption keys or data is a common response mechanism when device sensors or I/O detect common signatures of an attack or attempt to probe the FPGA or SoC for sensitive data.

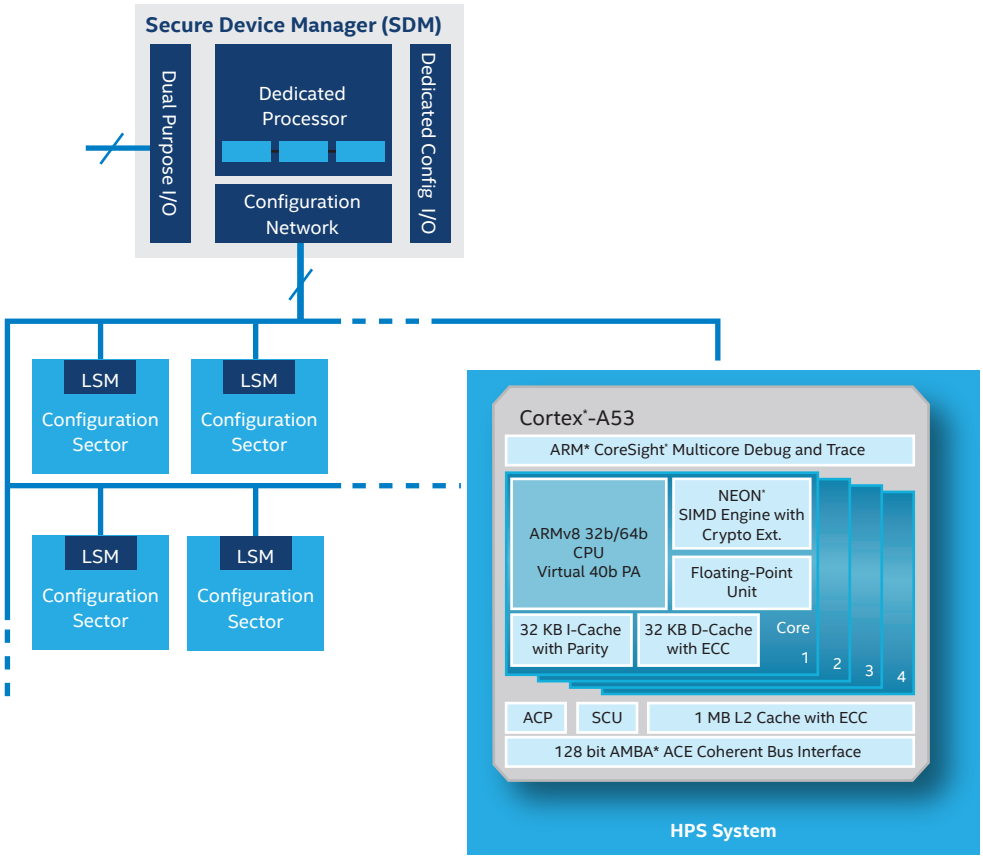

Figure 2. Configuration is Divided into Subsystems, and FPGA Fabric is Further Divided into Sectors

Because Intel Stratix 10 device configuration occurs by logical sector, overwriting, erasing, or zeroing configuration data can occur by logical sector. For highly sensitive or secure designs, zeroing by sector simplifies and reduces the zeroization time of a design if the sensitive information is isolated to, and contained within, a limited number of sectors. Designers may want to sanitize or zero the entire design anyway, but zeroing sectors can take place on a priority basis based on the sector sensitivity. The zeroization process (overwriting patterns and verification) are scripted in the SDM.

Configuration process

Intel Stratix 10 FPGAs and the SDM block provide the most robust, secure, and authenticated device configuration process in the industry. Some customization is also available in the configuration processes and IP protection for each user design. The degree of flexibility can be better understood by showing the high-level configuration data flow shown in Figure 3.

Figure 3 starts with a basic block diagram of the configuration data for an Intel Stratix 10 FPGA or SoC design. The configuration block data is the same whether in flash memory, SD card, or over a hard PCI Express connection (configuration via protocol). This configuration data is divided into several logical pieces, starting with the configuration data and code for the SDM itself. Other portions of the configuration data are logically divided into FPGA sectors and code blocks for the hard processor system in the Intel Stratix 10 SoC.

Loading and authenticating the configuration image

The first configuration step loads the SDM data. Because this stage manages all other security and keys for the Intel Stratix 10 FPGA, the SDM image is 100% authenticated against an Intel signature and verified with an on-chip Intel public encryption key. Other configuration options allow the designer to provide their own SDM image signature using a private encryption key and then verify this signature in configuration with a user-installed public encryption key.

Variables managed by the SDM

Based on designer decisions enabled in the Intel Quartus Prime software, the SDM decides how to ingest, process, and configure the remainder of the user design. These decisions include the configuration order of the Intel Stratix 10 device (e.g., FPGA first or HPS first, and the specific order of FPGA sector configuration). SDM instructions can also indicate thresholds and responses for environmental monitors. Licensable functions like the PUF are optionally included in SDM code based on Intel Quartus Prime software licenses. Finally, SDM instructions define the encryption key security by sector.

Authenticating, decrypting, and configuring by sector

A key, flexible feature of the SDM block for Intel Stratix 10 devices is the ability to make separate encryption and source encryption key decisions for the FPGA design on a logical sector basis. In this case, the designer can select different encryption keys for each sector or use a variety of encryption keys based on the sector sensitivity level. Different encryption key handling procedures can be designed for keys at different security or sensitivity levels. An encryption key can be used across multiple sectors or a single sector, to reduce the attack surface of that design sector. All encryption keys used for device decryption are protected by the device's root key.

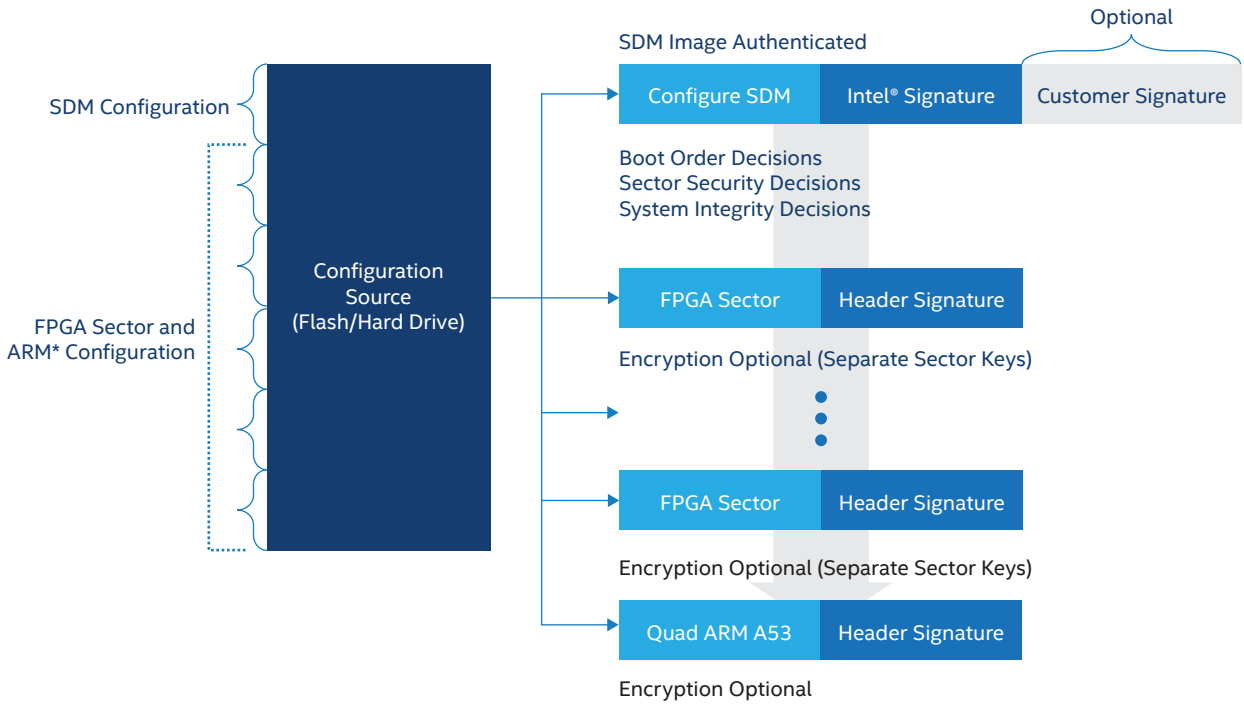

Figure 3. SDM for Intel Stratix 10 Devices Configures First, Followed by User-Designated Configuration Order

Addressing side channel leakage through programmability

One of the most common, documented attacks on existing programmable logic devices are side channel leaks on the configuration process, primarily targeting power. Intel Stratix 10 devices use a variety of methods to limit side channel leakage and attack surfaces. These methods include the pre-authentication of all sector-based data blocks by the SDM before encryption, dynamic encryption key updates, and encryption key diversity across sectors. The SDM itself is a key tool for reducing side channel leakage, as well as the many configuration methods available to the user. The configuration process can be specific to a design and can scramble or randomize the configuration data processing order. Additional techniques will be described in later user documentation.

The SDM controls the configuration scheme for each design. If a particular configuration process is found effective against the threat profile of your user environment, you can update the configuration process and re-authenticate it in the field.

Use case: multiple instance, multiple security level solutions

The SDM's unprecedented flexibility, and the ability to configure and secure logical sectors of the Intel Stratix 10 FPGA and SoC separately enables an important class of use cases for high-performance computing, data centers, and cloud-based environments.

As part of a large heterogeneous computing solution or data center, Intel Stratix 10 FPGAs and SoCs contribute high-performance hardware acceleration to a system solution. These accelerator functions can be highly diverse, and can use all of the resources of an Intel Stratix 10 device or only a small fraction of these resources.

When configuring a portion of the FPGA, the configuration data for that portion or sector can be encrypted separately and protected from the rest of that design. It can also be decrypted and authenticated by the SDM or other base user logic using different keys than the rest of the design. These encryption keys can be pre-loaded and protected into the Intel Stratix 10 device configuration, or provisioned to the device security in run-time. This multi-instance, multi-user security style makes Intel Stratix 10 FPGAs and SoCs ideal for environments where IP and sector-based designs are shared among multiple potential designers and users.

Robust, layered, and configurable

Over the 30-year plus history of configurable logic products, more and more sub-systems have added the moniker of programmable. These products include I/O and peripherals, soft and hard memory controllers, programmable DSP blocks, all the way up the multi-core embedded processors. One of the last frontiers of design in making the modern FPGA and SoC all programmable is the configuration process itself.

The advantages of the SDM and its programmability include faster configuration times, better and more diverse security, more robust and flexible SEU responses, and the ability to be updated. It enables secure authenticated device maintenance, debug, and key management functions. However, the most unique capability in the FPGA and SoC market is that the SDM diversifies and customizes the configuration process itself to address the most important classes of published attacks on programmable logic devices.

Where to Get More Information

For more information about Intel and Intel Stratix 10 FPGAs, visit **https://www.altera.com/products/fpga/stratix-series/stratix-10/overview.html**

[1] http://www.altera.com/en_US/pdfs/literature/wp/wp-01201-fpga-tri-gate-technology.pdf
[2] https://www.altera.com/documentation/jzw1474049428757.html#joc1432226473022
[3] http://www.intrinsic-id.com/wp-content/uploads/2014/09/PUF_aging.pdf
[4] http://www.intrinsic-id.com/wp-content/uploads/2014/09/SRAM-memories.pdf
[5] Jorge Guajardo, Sandeep Kumar, Geert-Jan Schrijen, and Pim Tuyls. FPGA Intrinsic PUFs and their use for IP Protection. Cryptographic Hardware and Embedded Systems - CHES 2007. Lecture Notes in Computer Science, Volume 4727, 2007, pp 63-80.
[6] Vincent van der Leest and Pim Tuyls. Anti-Counterfeiting with Hardware Intrinsic Security. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013, pp 1137 - 1142.
[7] Roel Maes. Physically Unclonable Functions: Constructions, properties and applications. Springer-Verlag Berlin Heidelberg, 2013.