Overview of Data Security Using Microsemi FPGAs and SoC FPGAs

November 2013

Introduction

The security of electronic systems can be divided into two major classes: Design Security and Data Security. The goal of Design Security is to ensure that the owner's hard work and valuable Intellectual Property (IP) are protected and intact at all times. Data Security refers to applications that a system utilizes to protect and authenticate data. Typical examples of Data Security would be protecting sensitive data transmissions (perhaps financial records) from snooping, authorizing a remote update (perhaps of embedded processor code) to insure that update is from a valid source, or making sure that a new hardware component added to a system (like an add-on card to a secure server) is from the authorized OEM. Applications for Data Security cut across every market segment and have become pervasive elements for Internet connectivity.

This overview will introduce several key Data Security concepts that are helpful in understanding the advanced features available in SmartFusion®2 and IGLOO®2 devices for implementing Data Security functions in embedded system designs. If you are unfamiliar with some of the security terms used in this paper it may be helpful to refer to the Online Security Glossary hosted on the Microsemi Security website. A link to the Glossary is given in the "References" section at the end of this paper.

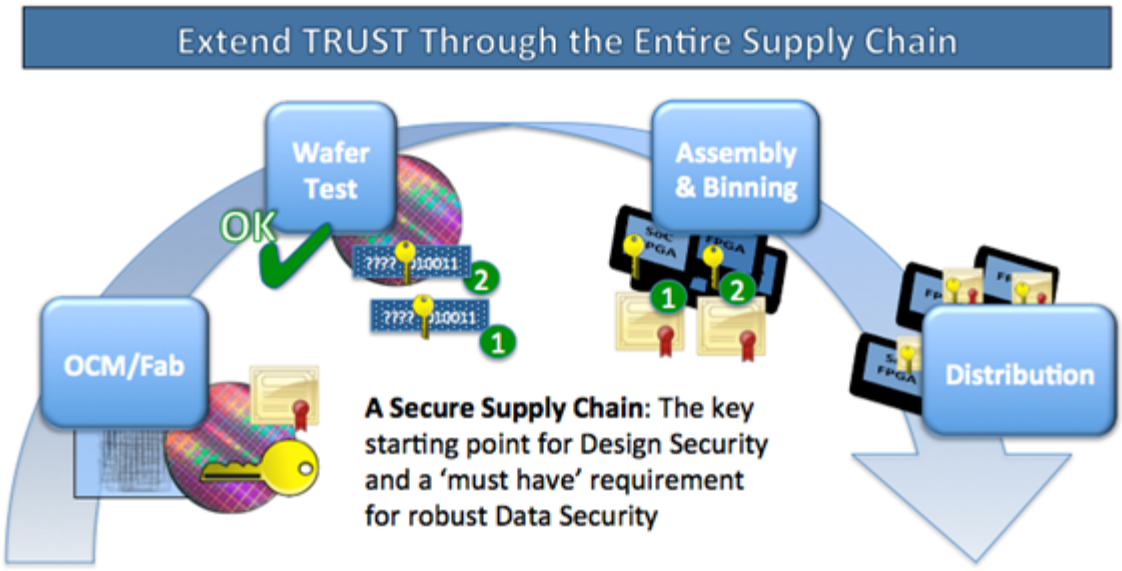

Figure 1: **Secure Supply Chain And Robust Design Security—The Key Ingredients for Successful Data Security**

The Starting Point for Robust Data Security

In order to successfully implement Data Security any design must first establish a secure supply chain so that purchase devices are free from the threats of counterfeiting and fraud. If a design is manufactured with faulty components it is impossible to guarantee Data Security functions will operate as required. As illustrated in Figure 1, Microsemi FPGAs and SoCs use flash technology and advanced cryptographic methods to build trust through the entire supply chain. Security keys, encrypted configuration bitstreams, and secure device IDs, using Certificates of Conformance, all contribute to creating a secure supply chain, the starting point for any secure embedded system. You can learn more about how this is accomplished in the secure supply chain related material listed in the "To Learn More" on page 10 at the end of this document.

Using devices, like SmartFusion2 and IGLOO2 devices, with robust supply chain protection from the threats of counterfeiting and forgery are a critical first step. Insuring that the target FPGAs and SoC FPGAs in your embedded design utilize robust Intellectual Property (IP) protection (like the advanced Design Security features found in SmartFusion2 and IGLOO2 devices) will insure the integrity of your underlying design, protecting it from attacks that could compromise the Data Security of your design. An understanding of Design Security is helpful when considering Data Security so it is recommended that the reader, if not already familiar with Design Security, read the paper or watch the Video on Design Security given in the "To Learn More" section at the end of this document.

The idea of a secure starting point is perhaps the most fundamental concept on which Data Security is based. The description of a Hardware Root-of-Trust will thus be the starting point for our overview of Data Security concepts.

Figure 2: **Hardware Root-of-Trust is Required for any Secure System**

Hardware Root of Trust

Software by itself is not secure, therefore fielded systems require a Hardware Root-of-Trust (RoT) as a secure starting point on which other security capabilities (either hardware or software) can be built. The Hardware RoT contains a protected process that always performs as intended and is secure from unauthorized changes, updates, snooping or other interference. Typically, a Hardware RoT consists of a security algorithm (usually implemented with one of a number of cryptographic standards) along with one or more secret ‘keys’ used to encrypt/decrypt or authenticate various security functions within the system. Once in place, the Hardware RoT can perform operations that extend the trust zone to cover other parts of the system, even allowing secure communication across an untrusted network. Some example functions a Hardware RoT could implement include:

- Signature checking of software stored in external memory
- Validation of system boards for authenticity/cloning
- Management of anti-tamper protection and application of penalties when tampering is detected

SmartFusion2 and IGLOO2 devices have all of the required security features needed to create a robust Hardware RoT. These devices are loaded with a secure Certificate of Conformance, which acts in part as a digital part number, by Microsemi during manufacturing.

This secure signature allows the purchaser to validate that the devices are exactly the same as those purchased, eliminating the possibility of forgery or counterfeit devices making their way into the design. Clearly a secure RoT must be built on a secure supply chain in order to be robust. SmartFusion2 and IGLOO2 devices also use an encrypted bitstream so that these devices, even if programmed at an unsecure contract manufacturer, are configured securely.

Once the underlying devices are known to be a secure ‘target’ for the RoT function it is critical that the devices contain the needed security key storage and cryptographic algorithms. The NVM-based storage mechanism used by SmartFusion2 and IGLOO2 devices is the most secure method of key storage available to FPGAs and SoC FPGAs. In particular, NVM-based devices are much more secure than SRAM-based devices, which store configuration data off-chip making it easy to capture the design and even reverse engineer the RoT design, opening it to a variety of security breaches. SmartFusion2 and IGLOO2 devices also contain a variety of additional features that support Security Key protection, security standards, and cryptographic algorithms required for a robust RoT implementation. These capabilities are explained in more detail in the following sections.

Security Key Protection

The first step in security key protection is to protect the initial key loading process. Microsemi uses a variety of techniques to insure this process is secure. More details on these techniques are available in the "Securing Your Supply Chain Life Cycle Video or White Paper" given in the "References" section at the end of this document.

Once the security key is successfully loaded into the programmable device, it is critical to protect the key from attempts to read it using various intrusive and non-intrusive forms of external attacks. For example, attempts to challenge the key can provide data in the form of response delay times or power spikes that provide clues as to the stored key values.

SmartFusion2 and IGLOO2 devices encrypt all security keys stored in on-chip NVM. This provides an additional level of security, increasing protection from possible attacks on security keys. Additionally, side-channel attacks, using Differential Power Analysis (DPA), can be mitigated by using Cryptographic™ Research Incorporated (CRI, now a division of Rambus) DPA-resistant technology.
Microsemi has obtained a license from CRI for the DPA patent portfolio, consisting of more than fifty patents. Microsemi is the only FPGA manufacturer licensed to use these patented forms of security keyprotection.

This license has two main components:

- It allows Microsemi to use any of CRI's patented techniques to protect the FPGA initial configuration and re-configuration process from side-channel attacks.
- It allows Microsemi to extend a sub-license to customers who purchase selected Microsemi FPGA “S”-Class devices from which the designer can then use any of CRI's patented DPA-mitigation techniques to protect their end-application from side-channel attacks. The protection techniques can be incorporated in the user's logic implemented in the FPGA fabric or in the user's firmware executing on a hard or soft microcontroller, in the licensed Microsemi FPGA. Example devices include:
  - M2S010T-FG484 – DPA-resistant Design Security
  - M2S010TS-FG484 – DPA-resistant Design Security plus CRI pass through license and access to cryptographic functions for data security applications

Many additional security key protection features are available through simple calls to the Security Subsystem. Just two of the many available features are the non-deterministic random bit generator (NRBG) service and User PUF key enrollment and regeneration service. Additional details on the various functions available via Security Subsystem Services is available in the "Security Subsystem Services" section later in this document.

Security Standards

The use of standard cryptographic algorithms is required when implementing Data Security capabilities. Standards are always the best choice when implementing Data Security, since it is important to be able to communicate securely with other systems which will be using standard security protocols. Standards support encryption/decryption using secret keys and/or authentication (the ability to prove a message is from a known identified source) using shared secrets. These two capabilities are hard requirements when creating a secure system. Some of the most common standards that typically need to be supported are SHA-256, AES-128/-256, and HMAC.

User Cryptographic Services (AES-128/256, SHA-256, HMAC-SHA-256, ECC, Pseudo-PUF, Key Tree)

- AES services: AES-128/256 encrypt/decrypt, ECB, CBC, CTR, and OFB modes
- Elliptic Curve Cryptography (ECC) services: point addition and scalar point multiplication on NIST P-384 curve
- "Pseudo-PUF" is a challenge-response service based on a 256-bit static per-device random secret stored in non-volatile memory
- Key Tree service derives an output key (or MAC tag) by successively hashing a root key (using SHA-256) 135 times in a DPA-resistant binary tree construction based upon a public 7-bit parameter and a 128-bit public input value

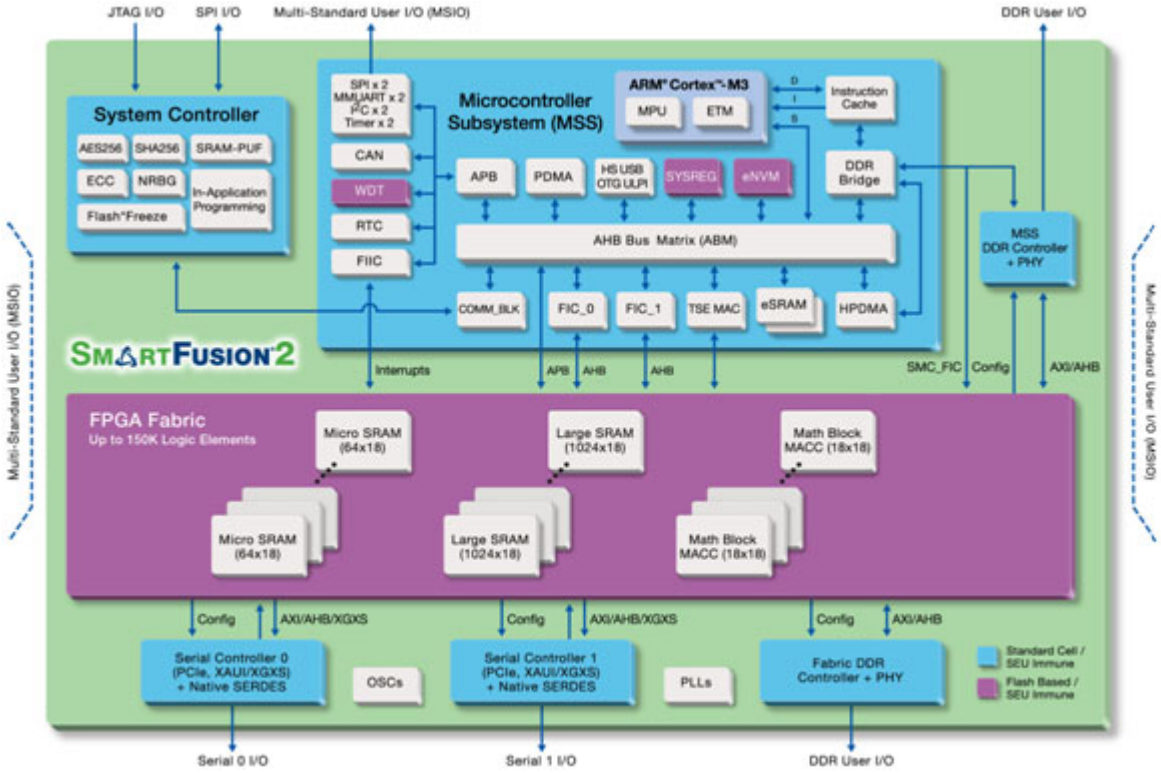

Figure 3: **Functional Block Diagram of a SmartFusion2 SoC FPGA**

Cryptographic Functions

There are additional cryptographic functions used to implement security key protection capabilities and that provide building blocks for higher level functions. One of the most advanced security key functions uses unique physical characteristics of the device to generate device unique secrets that can be used for authentication or encryption/decryption functions.

These unique functions act similarly to the person's unique 'biometric' characteristics of many security identification systems. Another common cryptographic building block is a true random bit generator that can be used as a seed for generating unique secrets or as a source for numbers used only once (a nonce). A nonce can be used to inject changes into an otherwise static data stream to improve security (For example, it can help eliminate a common attack where a previously sent encrypted message is resent as a 'replay' attack on a secure system).

SmartFusion2 and IGLOO2 devices support both a Non-Deterministic Random Bit Generator (NRBG) Service as well as an SRAM-PUF Key Enrollment and Regeneration Service to implement advanced security functions. The functions are easily called through Security System Services.

- Non-Deterministic Random Bit Generator (NRBG) Service (also known as the True Random Number Generator Service)
  - True random entropy source
  - Deterministic post-processor designed per NIST SP800-90 using AES CTR mode with 256-bit security strength
  - Hardened with additional entropy from pseudo-PUF or SRAM-PUF
  - Includes self-test and health monitors
- User SRAM-PUF Key Enrollment and Regeneration
  - Supports both "intrinsic" (self-generated random keys based on unique device "biometrics"), or "extrinsic" (imported keys up to 4096 bits)
  - Activation Code and Key Codes stored in dedicated private eNVM, or optionally off-chip
  - 256-bit security strength
  - Keys effectively disappear when power is removed from the dedicated SRAM
    Non-volatile security better than battery-backed SRAM, but without the battery

Microsemi provides a comprehensive suite of industry-standard data encryption Intellectual Property (IP) cores and partnerships that are available for SmartFusion2 and IGLOO2 designs. These IP cores extend security capabilities by allowing users to implement various functions in the programmable fabric. This additional level of customization makes it easy to support additional protocols and functional requirements not available from the 'hardened' device functions. For example, in some cases having multiple encryption blocks supports higher bandwidth capabilities. If multiple high-speed serial channels are used in the design, with each needing a decryption or authorization function, it might be more efficient to allocate separate decryption or authorization blocks for each channel, instead of trying to share a single block among all the channels.

The ever expanding Microsemi library of security IP functions includes functions such as:

- DES
- 3DES
- AES
- Pseudo Random Number Generators
- Secure Hash Algorithm
- RSA
- Elliptic Curve Cryptography
- GCM, tuned for 802.1ae

A number of the Microsemi hardware and firmware IP offerings are available with countermeasures to protect against security attacks, such as differential power analysis (DPA) through the license acquired by Microsemi from CRI.

This license allows Microsemi to extend a sub-license to customers who purchase selected Microsemi FPGA devices simplifying the use of DPA protected IP Cores.

Secure Remote Updates

One of the most useful features of a networked connection to an embedded system is the ability to remotely update code stored in an embedded processor or the configuration bitstream of an FPGA. Unfortunately, remote updates have recently come under increasing threat of malicious attacks and the theft of secure data through attacks on the remote update capability of embedded systems. A secure embedded system needs to implement a security 'barrier' around the update facility so that only authorized updates are recognized and acted upon. Additionally, remote updates should be protected from a common 'replay' attack where an older update is used to roll-back the version of the firmware to a previous, less protected version. Filtering out older, less protected versions, can be a critical element in securing a remote update capability and it is sometimes overlooked providing an 'attack vector' to malicious intruders.

SmartFusion2 and IGLOO2 devices provide the necessary underlying robust security capabilities needed to create higher-level functions, like a secure remote update. Several other key features are provided to simplify and further protect the remote update function. One of the most fundamental is the use of an encrypted bitstream. The bitstream used for configuration, and thus for remote updates, is encrypted and authorized so that attacks trying to use remote update (perhaps as a denial of service attack, where configuration bitstreams are sent repeatedly in an attempt to 'shut down' device) will be avoided. SmartFusion2 and IGLOO2 devices also accept a programming bitstream from a variety of communications interfaces such as SPI, USB, or Ethernet. This is of particular importance when creating a remote update function and dramatically simplifies the design of this capability.

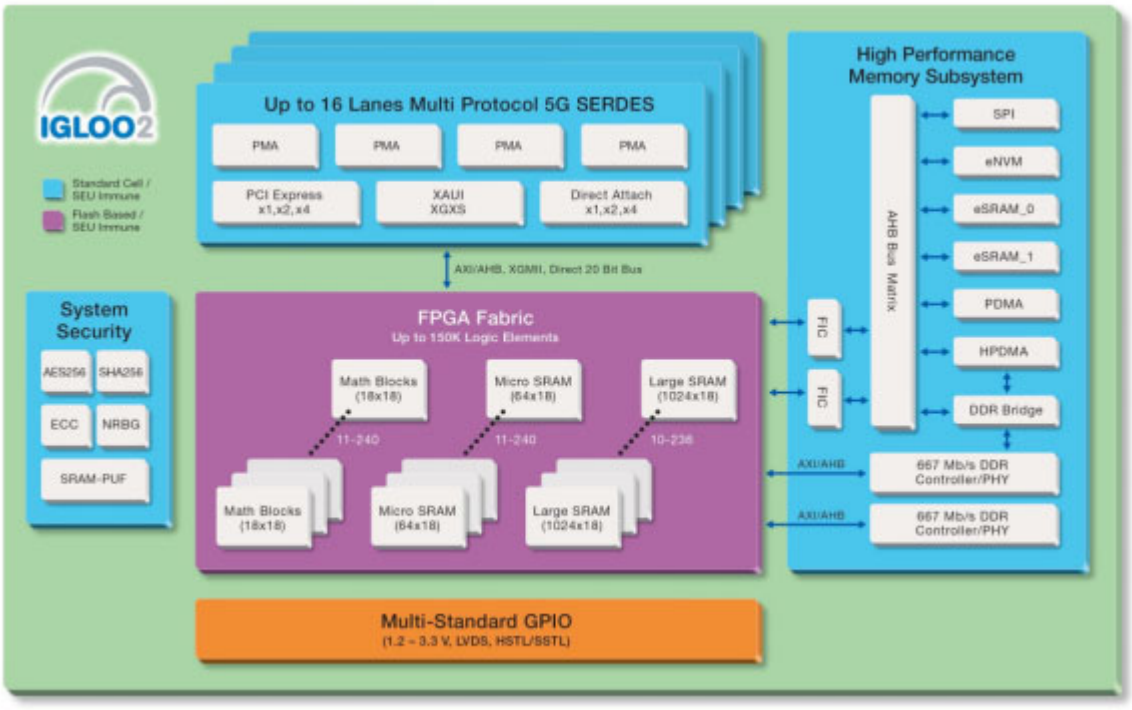

Figure 4: **IGLOO2 Functional Block Diagram**

SmartFusion2 and IGLOO2 devices, through the use of the System Security Subsystem, shown on the left of the functional block diagram of IGLOO2 in Figure 4, can implement programming autonomously.

This means that the hardware Root of Trust is used to implement the remote update, securing the operation from outside tampering. Autonomous programming supports several advanced capabilities including recover from a power down during programming. The controller records the event and keeps the device from restarting with an incomplete configuration. Programming can be restarted and only on a successful update is the device restarted. Building on these capabilities SmartFusion2 and IGLOO2 devices also support back-version protection. This eliminates the possibility of a ‘re-play’ attack where an older version of a configuration file is used to try and restore a ‘back-version’ configuration state with known security vulnerabilities. Additionally, a ‘golden’ configuration bitstream can be used to restore a know design version if, due to tampering, the current configuration is corrupted.

Secure Boot

Once a Hardware RoT is established and a secure remote update available, one could extend this zone of trust to securely boot a host processor. A secure boot process initializes an embedded processing system from rest. It does this by executing trusted code, free from any tampering by a malicious intruder. Without this level of trust an alternate boot image could replace the original boot code and allow an attacker to ‘hijack’ the entire embedded system. Just about any embedded system needs to be free from such attacks, for many embedded systems such an event could prove catastrophic. Secure system boot is typically a multi-stage process with the first stage using immutable trusted code to verify (using cryptographically secure algorithms and keys) the next stage code. The next stage can then be loaded and execution started knowing the code will operate as expected. This process continues for as many stages as required with each successive stage. At the end you have a secure boot process.

SmartFusion2 and IGLOO2 devices have all the capabilities required to create a robust secure boot. Building on the secure RoT previously described, a secure core function can be created to verify and load the initial code into the target processor, as illustrated in Figure 5 below. The immutable boot loader can be stored in the SmartFusion2 or IGLOO2 internal NVM and thus easily secured. The various cryptographic functions available through the Security Subsystem Service calls can be used to implement the needed authentication and any needed encryption/decryption functions (perhaps used if the balance of the code is too large for internal storage and must be stored external to the RoT in a serial flash device). Finally, tamper detection can be added with appropriate penalties being applied to protect the system from physical attacks. Refer to the white papers listed in the "References" section of this paper for more details on SmartFusion2 and IGLOO2 implementations of a secure boot.

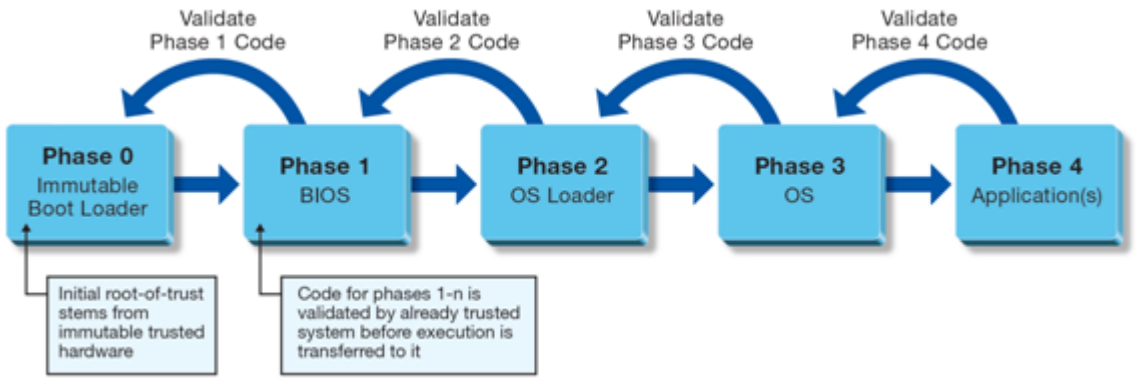

Figure 5: **Secure Boot Process Steps**

Security Subsystem Services

A multitude of security services is available through simple API calls to the Security Subsystem. These calls initiate or manage the hardware functions that ‘accelerate’ common cryptographic functions used to implement enhanced security capabilities. More details on several of these services and the methods used to manage them can be seen in the “SmartFusion2 and IGLOO2 Design Security and Cryptography Services” Webinar listed in the "References" section at the end of this paper. A list and short description of the major services is given below:

- AES-128/-256: Used for implementing encryption and decryption using the industry standard AES algorithm. Support for the most common operating modes- ECB, CTR, CBC, OFB is also available.
- SHA-256: Used for implementing message digests using the industry standard SHA algorithm.
- HMAC: Used to implement industry standard Message Authentication Code (MAC) algorithms. It is based on SHA-256.
- ECC: Elliptic Curve Cryptography functions for point multiplication and addition. Based on the industry standard NIST P-384 curve.
- KeyTree: Used for Key Derivation or MAC functions. This is a DPA-resistant alternative to HMAC.
- PUF Emulation: The Physically Unclonable Function (PUF) emulation service provides a challenge/response-like protocol that is either based on an SRAM-PUF (in larger device) or a Pseudo-PUF implementation in smaller devices.
- SRAM-PUF: This service provides key enrollment and deletion, activation code management and key regeneration functions—all needed for SRAM-PUF key management. Intrinsic (internal) and extrinsic (external) design or data security keys are supported.

Additional Hardware Security Capabilities

The ability to build ‘barriers’ between different hardware functions on-chip can be very helpful in isolating secret information from unwanted scrutiny or snooping. Many forms of attack use design ‘bugs’ and the unintended consequences of firmware ‘corner cases’ which can unknowingly expose otherwise secure information. If on-chip security ‘locks’ can be used to limit hardware access (even in the event of a software/firmware bug or system error) the underlying security of the application can be vastly improved. For example, hardware firewalls that restrict software/firmware access to various protect memory pages is a common on-chip hardware technique for securing sensitive data.

SmartFusion2 and IGLOO2 devices provide an extensive set of security features that can create security ‘barriers’ and checks for various hardware elements in your design. For example, all segments of NVM can be checked for integrity by using a security service to generate a secure ‘hash’ on the contents and compare it to a previously generated (and protected) hash value. This checks that the memory contents have not been corrupted or compromised by an intruder. A variety of User Lock-Bits are available to lock-out access of specific functions. For example, the Debug port can be set to full access, no access or observe only access to eliminate the possibility of an attacker using the Debug port maliciously. Lastly, the Hardware Firewall function described earlier is available in SmartFusion2 and IGLOO2 devices and some of the detailed operational characteristics are given below:

- Block level hardware read-write protection based on the class of bus master accessing the memory
  - Up to four 4KB sectors with an independent additional layer of hardware controls
- Page-level hardware write-access control (that is, individual 1kbit pages can be declared as ROM)
  - Page-level software write-access control to prevent accidental overwrite

Anti-Tamper Protection and Penalties

Embedded systems can be subject to a variety of malicious attacks and many times attacks can be physical in nature, using logic probes, signal analysis and even board damaging trace cutting or drilling. These physical attempts to isolate key design modules can be an effective means of attacking a system and are common when a system is available, within a lab environment, for such aggressive scrutiny. It can be important to protect embedded systems from this level of threat so it is common to add ant-tamper protection features to the design that can check for physical attacks on the system. Some anti-tamper approaches use physical containment and can detect when a ‘seal’ has been broken. Other, lower cost techniques add extra traces on the board to form a protective mesh. If the mesh is disturbed in an unexpected way (or disruptions are repeated too often) penalties can be applied to prevent the successful acquisition of secret information. A penalty might be as simple as resetting the system or as complex as erasing all secured information from the system.

SmartFusion2 and IGLOO2 devices can provide advanced anti-tamper protection capabilities and penalties to simplify this level of protection for your designs. The use of digests for memory segments, where data is stored within SmartFusion2 and IGLOO2 devices, makes it difficult for an intruder to tamper with internal data. The use of DPA-resistant cryptographic algorithms further protects a device from tampering attacks. Additional anti-tamper techniques that use redundancy at the circuit level can also detect if changes to secure data are made in an unauthorized manner. Finally, the wealth of I/Os available on FPGAs can easily be used to create signal meshes as low cost tamper detection elements.

Both SmartFusion2 and IGLOO2 devices can also use a zeroization service to apply a variety of penalties that erase (‘zeroize’) secure data. Less stringent zeroization penalties can be used to just erase secure data while the most stringent can actually completely disable the device, making it impossible to even reprogram the device into a new working state.

Conclusion

This white paper has provided an overview of several of the most important Data Security concepts in the context of implementations using SmartFusion2 and IGLOO2 devices. These concepts should provide the reader with an excellent starting point for digging deeper into the security needs for a specific design. Microsemi provides a wealth of detailed information for learning about and implementing the most common security requirements. To find out more visit the Microsemi Security website or explore the "To Learn More" material listed below.

References

Online Security Glossary

To Learn More

1. Securing Your Supply Chain Life Cycle Video or White Paper
2. Overview of Design Security using Microsemi FPGAs and SoC FPGAs
3. How Easy is it to Secure Your Designs? Video
4. What is Design Security in a Mainstream SoC? Chalk Talk
5. SmartFusion2 and IGLOO2 Design Security and Cryptography Services Webinar
6. Overview of Secure Boot with Microsemi IGLOO2 FPGAs
7. Overview of Secure Boot with Microsemi SmartFusion2 SoC FPGAs

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo CA 92656 USA
Within the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

Microsemi Corporation (NASDAQ: MSCC) offers a comprehensive portfolio of semiconductor solutions for: aerospace, defense and security; enterprise and communications; and industrial and alternative energy markets. Products include high-performance, high-reliability analog and RF devices, mixed signal and RF integrated circuits, customizable SoCs, FPGAs, and complete subsystems. Microsemi is headquartered in Aliso Viejo, Calif. Learn more at **www.microsemi.com**.

55900162-0/11.13