Design of Secure Smart Energy Metering and Control System

Application Example

Introduction

Embedded Networked Systems control an ever-increasing percentage of the modern industrial infrastructure. Smart energy grid installations, complex chemical processing and transport facilities, the multiple modes of our transportation infrastructure, as well as storage/access systems for personal medical and financial information all use complex embedded systems that require advanced security features to prevent malicious users from system intrusion. Standard approaches to protecting sensitive embedded systems have evolved over the last couple of decades providing the critical algorithms needed to secure sensitive data and embedded processing functions. The recent development of integrated devices with security specific features makes it possible to now protect sensitive networked embedded systems from aggressive intrusion without sacrificing system performance, power cost, or board space.

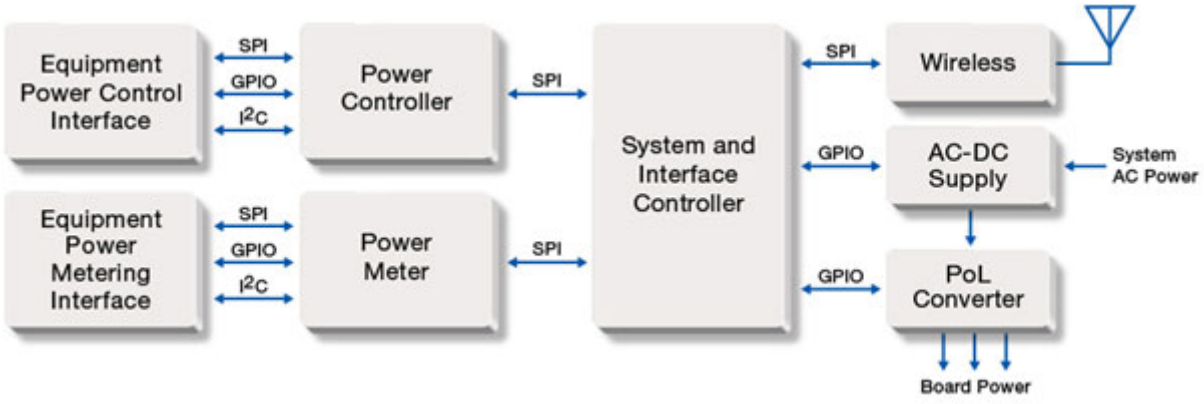

Figure 1: **Example of Smart Energy Meter and Control System**

A good illustrative design example makes it easier to identify security requirements and implementation options for networked embedded systems. A pervasive element of the Smart Grid energy management and distribution system is the Smart Energy Meter and Controller, a block diagram of a typical system is shown in Figure 1. These devices measure energy consumption at the customer (or even equipment level in large installations); so that the distribution control system can determine energy use requirements. Energy delivery can then be optimized, through the attached Smart Energy Controller, based on current and predicted requirements taking into account factors like weather, time of day, day of week, or even building occupancy levels. A networked system is necessary to transmit and receive measured data and control information. Unfortunately, these types of networked embedded systems can also be the targets of malicious hackers (using advanced attacks like the so-called Stuxnet computer worm).

The system and interface controller manages the entire Smart Energy Meter and Controller system and communicates to the Energy System Aggregator (a communications hub for the Smart Energy Network) through a wireless module. The system controller manages the system power supply to monitor and regulate power to the system board. It communicates through a SPI bus to the power controller, which modulates power delivery to the variety of equipment in the installation. Typical equipment might include heating, ventilation and air conditioning, lighting, process control, or even equipment racks. A SPI bus is also used to communicate to the power meter that monitors power use, building sensors, and environmental conditions. This information is used by the local controller to manage power but is also sent to a central energy grid controller to help optimize energy delivery for the entire grid.

Security Requirements

Our example design has several key security requirements to protect against aggressive intrusion. Much of the data transmitted and received over the network can be sensitive and should be protected from snooping or other interference. Also, the ability to update the code running in the Smart Energy Controller via the network is a useful feature but one that can be used by hackers to gain control of the system. A secure boot capability, where MCU code is validated prior to boot so that any changes to program code that are attempted by hackers is detected and corrected, will also be required if we are using an MCU to implement one or more of our controllers.

It is clear that we will require secure data transmissions, perhaps through standard security keys, to keep unauthorized users from observing or interfering with network data used by the system. Cryptographic keys must be protected and managed to keep them secure so that intruders can't determine the keys through sophisticated attack methods; even when an example system can be obtained and subjected to board and device level measurements.

In most cases the actual design must be secure as well or an attacker could 'reverse engineer' the design to obtain information that could be used to spawn successful attacks. We must have robust Design Security features that protect our design by preventing design changes (insertion of Trojan Horses, for example), and controlling the number of copies made throughout the device life cycle. For example, tamper protection will be a useful feature to minimize an attacker's ability to successfully gain access to secure design data, even if the gain possession is of an example system. We must secure the design from initial production, including any in-the-field upgrades, and even decommissioning of the design at the end of its life. Without a secure design it is difficult to adequately protect secure data.

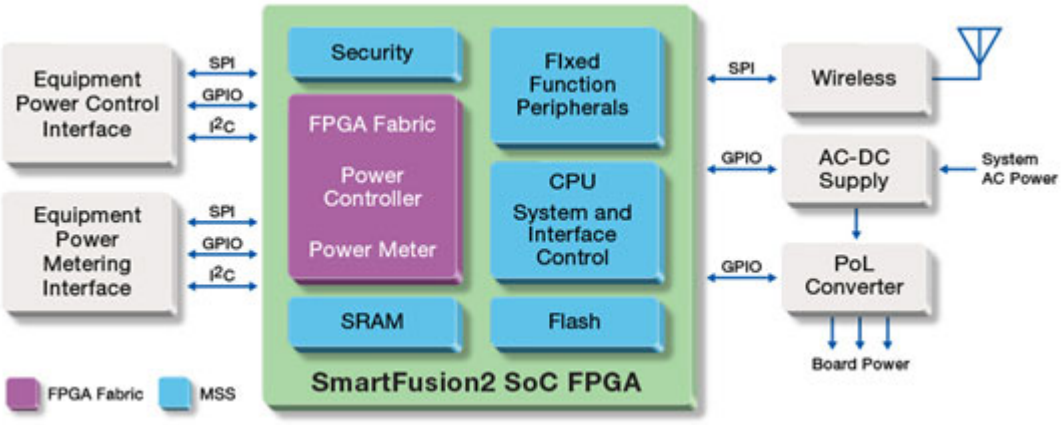

Figure 2: **Example of Implementation with Added Security Features**

Implementing Security Features in the Example Design

Let's take our example design and look at how we can significantly improve security by addressing the security requirements previously described. Figure 2 on page 2 shows an implementation example using the Microsemi SmartFusion®2 SoC FPGA. In this implementation we will combine the three controllers into a single device. The System and Interface Controller can use the hardened MCU within the SmartFusion2 device, (a part of the Microcontroller Subsystem, or MSS, key components of which are identified in blue in the figure), while the other two controllers can be implemented with the FPGA fabric (identified in purple in the figure) using a few state machines along with I^2C and SPI IP Cores. A single device implementation (since the SmartFusion2 device requires no external configuration device it is a true single chip solution) simplifies security considerably.

With all the control functions embedded within a single device, only one set of keys and associated algorithms are required to protect data inside the device or transmitted to/from the device. Design protection is also much simpler since only a single device needs to be protected from reverse engineering, cloning, overbuilding, tampering and other similar attacks on the actual design IP.

SmartFusion2 devices use on-chip Nonvolatile Memory (NVM) storage technology to configure the programmable logic elements in the FPGA fabric and the code and data storage used for the MCU. This technology makes it very difficult to reverse engineer the secret settings and security keys critical to establishing a starting point for any security algorithm. Additionally, the use of encrypted bitstreams for device programming and cryptographically robust security key loading algorithms makes it impossible for an attacker to determine secret data, even if they have access to the manufacturing facility where devices are personalized prior to being inserted in the target system. Thus, SmartFusion2 devices can be trusted to provide your embedded system with a secure starting point for your design.

Because SmartFusion2 devices are so secure, they can act as the Hardware Root-of-Trust (RoT) critical to implementing secure systems. A Hardware RoT is the most secure portion of the system that stores all security keys, and implements all the security algorithms. The Hardware RoT can then be used to extend the zone of trust to cover other parts of the system, even allowing secure communications across an entire entrusted network. Examples of these types of zones include the execution of secure boot code, signature checking of software stored in external memory, and validation of system boards for authenticity to combat cloning. For more details on Root-of-Trust and secure boot refer to the material listed in the "To Learn More" section at the end of this paper.

Support for Autonomous Remote Updates

Perhaps one of the most important features to keep secure is the facility to remotely update the code stored in the MCU or the FPGA fabric. Since the SmartFusion2 devices use an encrypted bitstream to program configuration memory, this data can be conveniently sent over the network without any risk of the configuration data being observed, or maliciously interfered with. The insertion of a Trojan horse in the configuration code, which could occur if the data isn't encrypted, would allow an attacker to take over the entire system, with potentially catastrophic results. SmartFusion2 devices support remote updates with a variety of advanced features. Autonomous programming allows the SmartFusion2 device to reconfigure itself, since the on-chip MCU and various communications ports can operate independently of the FPGA fabric. Additionally, the remote update process is protected from attempts to maliciously replay a previous update, in the hopes of re-introducing a 'back-revision' state with a know-security breach. Additionally, the SmartFusion2 updater can also use a 'golden code version' to restore a know-good state to the system if errors or tampering is detected. This provides another layer of security to make remote updates even more robust.

Using Microsemi SmartFusion2 SoC FPGAs

A block diagram of the SmartFusion2 SoC FPGA is shown in Figure 3 on page 4. The hardened CPU and associated peripherals are shown in the large blue block at the top of the diagram. The system controller, shown in the smaller blue block at the top left, includes a range of key security related functions that help with our implementation.

The FPGA logic is shown in the large purple block in the middle of the diagram.

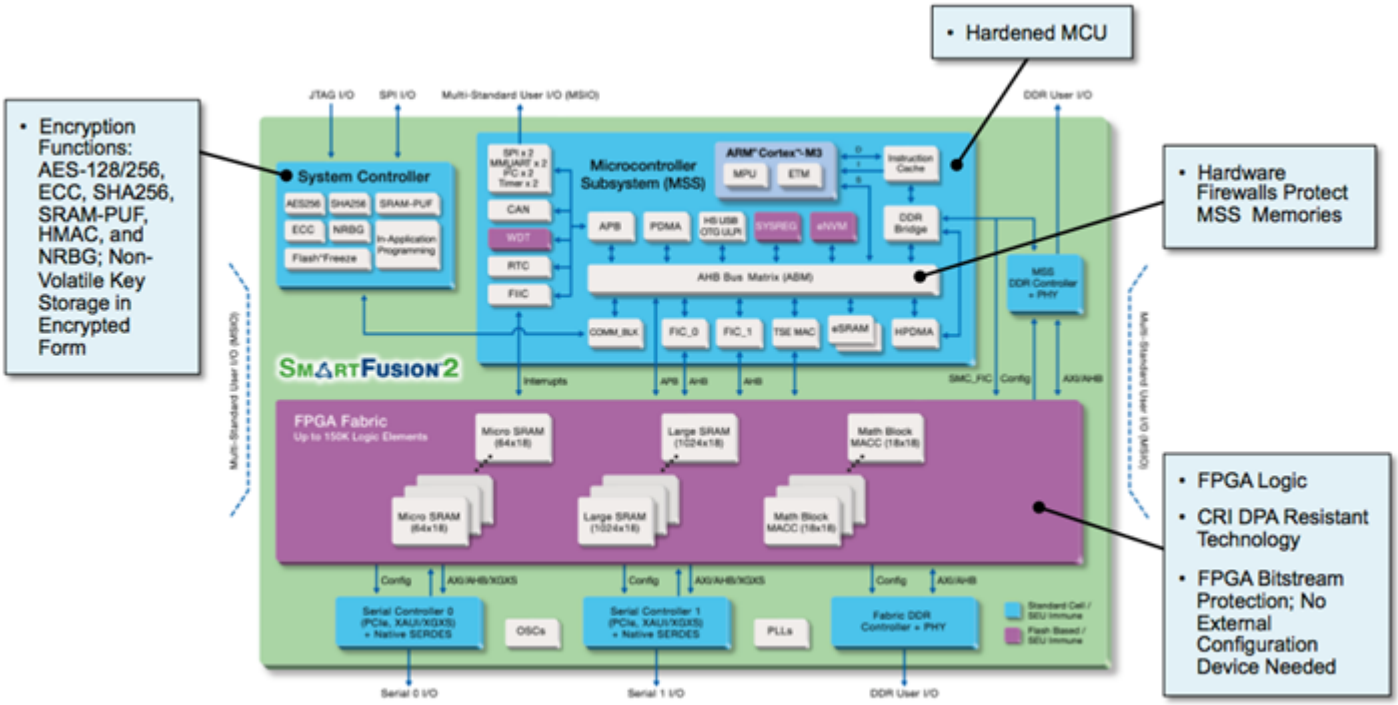

Figure 3: **SmartFusion2 Architectural Block Diagram and Key Security Features**

Many of the security requirements in our example design can use the security features in the SmartFusion2 devices. The requirement for using standard encryption algorithms to protect data transmitted and received within the system can be addressed through the broad set of encryption algorithms supported on SmartFusion2 devices. The algorithms include DES, 3DES, AES, Pseudo Random Number Generators, Secure Hash Algorithm, RSA, Elliptic Curve Cryptography (ECC), and GCM for 802.1ae. Several of these functions are available as hardened IP blocks and can be used as hardware accelerators to simplify the creation of a target application with security features of its own. For example, AES-256, SHA-256, and HMAC cryptographic services are all available to the target application. Additionally, a 384-bit Elliptical Curve Cryptography (ECC) engine is also available to further secure application data. A Pseudo-PUF challenge-response service is available to help establish application level key exchanges needed to establish trusted connections. SmartFusion2 devices thus not only establish a secure environment for the target design they also assist in the creation of any security functions needed within the entire system.

The requirement to further protect the design data from intrusion can also be supported by many of the advanced features of SmartFusion2 devices. For example, the use of flash technology with trusted encrypted user bitstream key loading during device programming protects the design data against supply chain attacks like cloning, overbuilding, reverse engineering, and counterfeiting.

Devices can be programmed by contract manufacturing houses, but the design data can be secured and only the required number of units produced by using SmartFusion2 on-chip security keys and encrypted bit-streams. Even more sophisticated attacks on design data are evolving and additional features are required to address these new developments.

Protecting Against Advanced Threats

Side-Channel Analysis is one of the most sophisticated forms of attack on cryptographic systems that uses information that leaks, unintentionally, from the real-world implementations of cryptographic hardware. For example, an attack might examine the characteristics of a cryptographic device when a variety of security keys are presented. Even if the keys are incorrect measurements and analysis of the power use (called Differential Power Analysis, or DPA), timing responses or electromagnetic radiation given off could provide clues as to the nature of the protected keys or algorithms used within the hardware RoT described earlier. An example of such an attack is shown in Figure 4 below, where the stored password can be determined by observing the current used when different passwords are presented to the device under attack. In this example, the activity of the multiplier block is easily determined and when the detailed observations are done, as shown in the blow-up at the bottom of the figure, the values for the password at each operation cycle can be deduced.

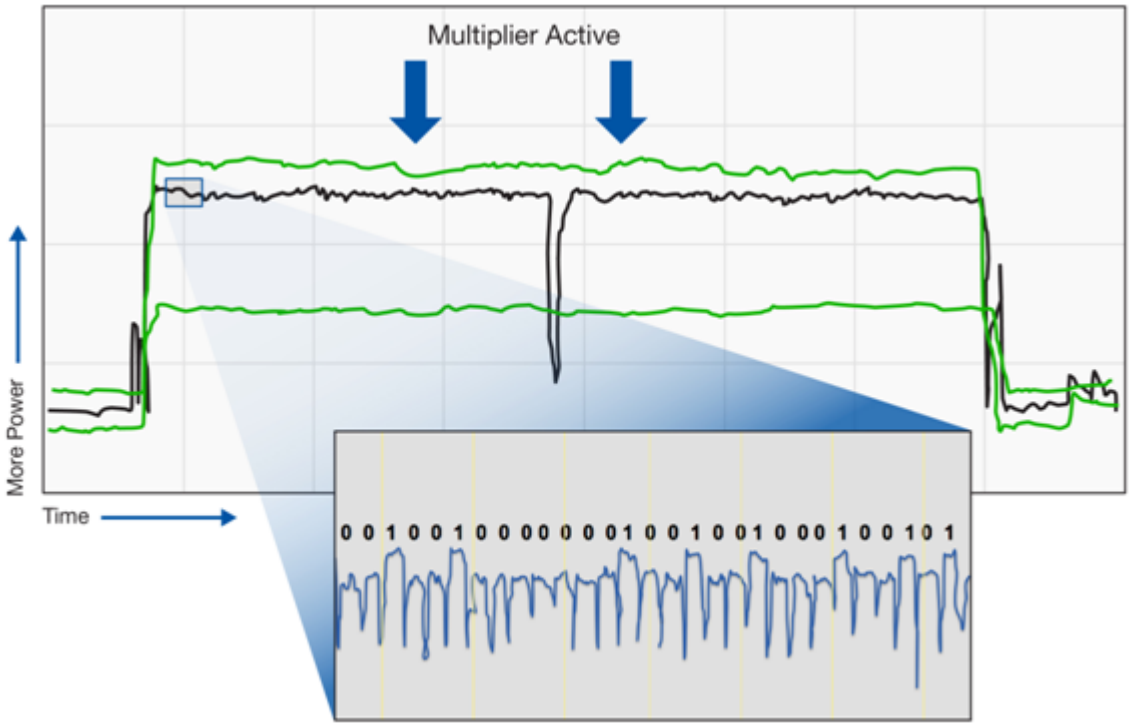

Figure 4: **Example DPA Attack on Stored Passwords**

Thus, given enough time and response examples it may be possible to ‘crack’ the stored keys and gain access to the other zones of trust for malicious purposes. SmartFusion2 devices include special patented DPA functions, licensed from Cryptography™ Research, Inc. (CRI, now a division of Rambus), that can protect against side-channel attacks to the FPGAs initial configuration, and re-configuration process as well as any active encryption/decryption data processes. For more details on side-channel analysis and DPA in particular refer to the material listed in the "To Learn More" section at the end of this paper.

Tamper Detection and Penalties

Hardware tamper detection is another way of determining if the design is under attack. Many times an attacker will cut traces on the board, in an attempt to isolate devices to implement hardware-based attacks.

One simple method of detecting such attacks is to use extra I/O signals to route traces throughout the board so that any attempts to cut or drill the board can be easily detected. SmartFusion2 devices can easily use FPGA logic to implement tamper detection circuits.

If an attack is detected it is important to take action, commonly called a penalty, to mitigate any damage an attack might generate. Zeroization is the practice of erasing sensitive parameters to prevent their disclosure if the system is attacked, or is at an increased risk of unauthorized access. SmartFusion2 devices support three different zeroization options, in addition to the ability to disable zeroization. The 'Like New' option retains factory keys and serial numbers and the device can be reprogrammed, if desired. The 'Recoverable' option zeroizes factory keys, but new factory keys and serial number can be injected using authenticated key recovery service. The 'Unrecoverable' option zeroizes everything and locks the device from all further operations. These capabilities make it exceedingly difficult for a detected attack on a SmartFusion2 device to capture any sensitive data prior to erasure.

Conclusion

Networked embedded equipment, like Smart Energy Meters and Controllers found throughout the Smart Energy Grid need to be protected from a variety of security threats to avoid the possibility of significant financial losses due. Threats to the design of the embedded system must be defeated by robust Design Security capabilities, while threats to the confidential data stored or transmitted within the system must be defended by strong Data Security features. Microsemi SmartFusion2 SoC FPGAs provide a host of advanced features to support robust implementations of both Design Security and Data Security. These features include the use of NVM for storing configuration data, encrypted bitstream programming, DPA resistance to protect security keys and cryptographic algorithms, advanced hardware tamper detection and penalty features, and the ability to create a robust root-of-trust on which more advanced features like autonomous remote updates and Data Security capabilities can be created. SmartFusion2 SoC FPGAs are the ideal target for securing your embedded systems designs.

To Learn More

Side-Channel Analysis and DPA

1. Protect FPGAs from Power Analysis
2. How Easy is it to Secure Your Designs?
3. What is Design Security in a Mainstream SoC Chalk Talk

Root-of-Trust and Secure Boot

1. Overview of Secure Boot with Microsemi IGLOO2 FPGAs
2. 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**.

55900159-0/11.13