Design of a Safe and Secure Process Control System

Application Example

Introduction

Embedded Networked Systems are increasingly called upon to control vast sections of the industrial infrastructure in the modern economy. Some systems require extraordinary safety and reliability to eliminate, as much as possible, failures that can result in dramatic financial losses or the loss of life. Familiar examples of these safety critical applications are mass transportation, power generation, and oil drilling/transport. Embedded systems are also used in applications where the results of failures are not catastrophic, but can still result in significant losses in process or manufacturing efficiency. When faults are detected and failures avoided significant material losses or manufacturing efficiency losses can be prevented. Additionally, a networked system is not really safe if it is not secure. Malicious users can hijack an embedded system or an embedded system can become the (perhaps unintentional) target of a virus or worm. These types of attacks can damage or render inoperable an entire system or complex. Clearly in many cases, both advanced reliability and security capabilities will be requirements in networked embedded designs.

This application example will identify the requirements for creating a safe and secure embedded process control system and show specific implementation techniques using Microsemi SmartFusion®2 SoC FPGAs. Some of the key topics covered include: the IEC61508 Standard and Safety Integrity Levels; Single Event Upset as a source of errors; the concepts of Dual Modular Redundancy (DMR), Triple Modular Redundancy (TMR), and design diversity; the reliability advantages of Zero FIT (Failure in Time) rate flash-based FPGAs; securing the supply chain and manufacturing process from counterfeit and fraudulent devices; securing the design from reverse engineering or tampering; securing the remote update process from malicious attacks..

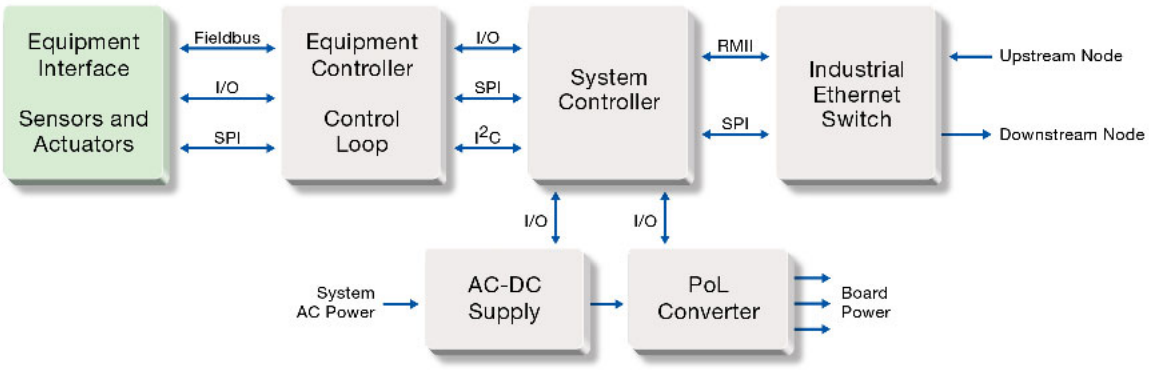

Figure 1: **Example of Embedded Network Process Control System**

Perhaps looking at an example design can best illustrate some of the key aspects and implementation options when improved reliability and security are required. Process control systems are one of the most useful examples to consider, particularly since the discovery of network transmitted worms that attack not only traditional PC operating systems, but embedded control systems as well (like the so-called Stuxnet computer worm). A block diagram of an example embedded process control system is shown in Figure 1 on page 1.

In this design, an Industrial Ethernet Switch is used to connect the controller to the network through an upstream node and a downstream node. A System Controller manages the overall operation of the Process Control System, including the Ethernet Switch and the power subsystem. A separate Equipment Controller, supervised by the System Controller, manages the equipment interface. The Equipment Controller implements any low level control loop processes required by the system. Higher-level process management resides within the System Controller under supervision through the network, perhaps by a centralized system that manages the entire manufacturing or chemical processing complex. This separation of control functions simplifies the implementation of the real time aspects of both the equipment control and network traffic management (For example, interrupt response time, memory bandwidth allocation, and active task priority determination). Let's look at ways to make this example system more reliable and secure.

Safety Requirements for the Example Design

To craft a sufficiently safe embedded systems, several key requirements need to be satisfied. First of all, the safety level appropriate for the application needs to be identified. The IEC 61508 standard is very helpful in determining the importance of safety in a particular application. Once the safety level is identified, common sources of failures need to be explored to insure all applicable sources have been taken into account. In our example design, we will look at some typical sources (like transmission errors and single event upset events) to show how we can use some common mitigation design techniques to reduce these errors to an acceptable level. (The IEC 61508 standard document identifies many more possible error sources and it is useful to read it over before getting too far along on your design. The techniques shown in this paper are appropriate for a wide range of error sources however, so we need not look at more than a couple to illustrate how the techniques can be applied more generally).

IEC 61508 and Safety Integrity Levels

All systems will have the possibility of failing, since it is impossible to design a system with an absolute zero failure rate. Thus each application should be designed with a target acceptable failure rate level.

As shown in Table 1, the IEC 61508 standard specifies acceptable failure rates for a variety of Safety Integrity Levels (SILs), based on the consequences of a system failure. The specification originally applied is solely at the system level but has also been applied to product and components by addressing Electrical, Electronic, and Programmable Electronics for both hardware and software. We will assume that our design falls within SIL Level 2 (perhaps because the controller manages a hazardous liquid as part of its function).

Table 1: **IEC 61508 Safety Integrity Levels**

SIL Level	Probability of Failure	Consequence	Application Example
4	1 failure in 110,000 years	Potential for fatalities in the community	Nuclear Power Plant Control
3	1 failure in 11,100 years	Potential for multiple on-site fatalities	Hazardous area laser curtain sensors
2	1 failure in 1,100 years	Potential for major on-site injuries or fatalities	Hazardous liquid flow meter
1	1 failure in 110 years	Potential for minor on-site injuries	Thermal Meter

Looking at the example design shown in Figure 1 on page 1, we can imagine some possible failure modes and their effect on the overall system. An error in the Equipment Controller might allow hazardous liquid to build-up in the system until a rupture takes place, creating a life threatening system failure. Similarly an error in the System Controller might miss warnings from the Equipment Controller, that could also result in life threatening failures. An error in the Ethernet Switch (a constant message broadcast for example) could bring down the entire network and threaten the entire complex, not just a single node. Note that the System Controller also manages the power supply subsystem, (not an unusual feature of embedded controllers) so an error associated with the power supply could cause a dramatic system failure. This is also a potential weakness for a malicious attacker to exploit if they wanted to inflict permanent damage on the system.

Protection from Transmission Errors

We also need to look at possible failure modes when remote code updates or other sensitive messages are sent over the network. Without a sufficient level of data protection, transmission errors or malicious attacks could alter program code execution, incorrectly adjust trigger levels, or capture sensitive operating parameters. Standard error detection functions (like a Cyclical Redundancy Check or CRC) can be used to protect messages from transmission errors. The Ethernet Switch will automatically check messages for errors using this technique. If required, the System Controller can implement additional Error Detection and Correction functions. Cryptographic protocols and standard encryption algorithms can be used to improve the security of network traffic within the system by securing the data in transit and authenticating remote facilities. These requirements and the associated design techniques will be covered in more detail in the "Security Requirements for the Example Design" section later in this document.

Single Event Upset (SEU) as a Source of Errors

The Single Event Upset phenomenon was first discovered in 1979 by Intel and Bell Labs as failures in DRAMs and is attributed to stray alpha particles or neutrons ‘flipping’ the memory cell. In 1999, Sun Microsystems noticed errors in cached SRAMs for mission critical servers. In space and aviation applications, the effects of radiation on electronics is well understood as operational altitudes have a higher neutron flux. However, the SEU phenomenon is increasingly becoming a concern at sea level as well. The continuous drive to smaller semiconductor geometries reduces the charge at each SRAM cell and the ever increasing content of electronics in fielded systems increases the likelihood of SEU related SRAM errors. Note that flash memories, which require a significantly higher energy level to ‘flip’ state, are immune to these types of SEU events.

Mitigation of Errors Through Redundancy and Design Diversity

In safety critical systems redundancy is mandatory to operate properly in the event of a failure. There are two well-known techniques that are widely utilized—Dual Modular Redundancy (DMR) and Triple Modular Redundancy (TMR). In the case of Dual Modular Redundancy, duplicate designs work in parallel. Each processing element receives the same input and a fail-safe certification engine checks for consistency. If a fault is identified, then prevention must be taken to avoid a failure. Triple modular redundancy creates three duplicate designs and the results of each output are presented to a voting circuit, such that the output state that receives the most votes is set. This can withstand the complete failure of one subsystem and allows a supervisor circuit to attempt to fix the fault, or alert an operator.

A design diversity methodology is sometimes employed to further improve reliability. Using this methodology, parallel designs are not just duplicated but will perform the same function using a different implementation. For example, an FPGA might be used for one of the designs and the parallel design might use an MCU. This diversity in the target implementations increases reliability even more since errors related to complex design or implementation ‘bugs’ will not be duplicated in dramatically different targets.

Implementing Redundancy and Design Diversity in the Example Design

Let's take our example design and look at how we can significantly improve reliability by using the redundancy techniques previously described. Figure 2 shows the changes to the example system.

In order to improve network reliability we added redundant Ethernet connections to the upstream node and the downstream node. The new redundant power subsystem helps recover from a failure in the main supply. Power will switch over to a redundant supply if the main supply fails. The System Controller and Equipment Controller are now each implemented using a Dual Modular Redundancy (DMR) technique, as illustrated in the 'blow-up' of the Equipment Controller (The System Controller would use a similar technique). The controller functions are duplicated and compare logic is added to identify any outputs that do not 'agree'. When such an error is detected, the subsystem responsible for the error can be reset and diagnostics performed. This mitigates the chance of the error resulting in a system failure. Note that the dual implementations of the System Controller use a design diversity technique. One controller is implemented with an MCU and the other is implemented with an FPGA. This provides additional reliability, since each implementations error characteristics will be significantly different and thus the chance of a common systematic error (for example, their response to noise, temperature, voltage, timing differences, or even implementation 'bugs') will be significantly reduced.

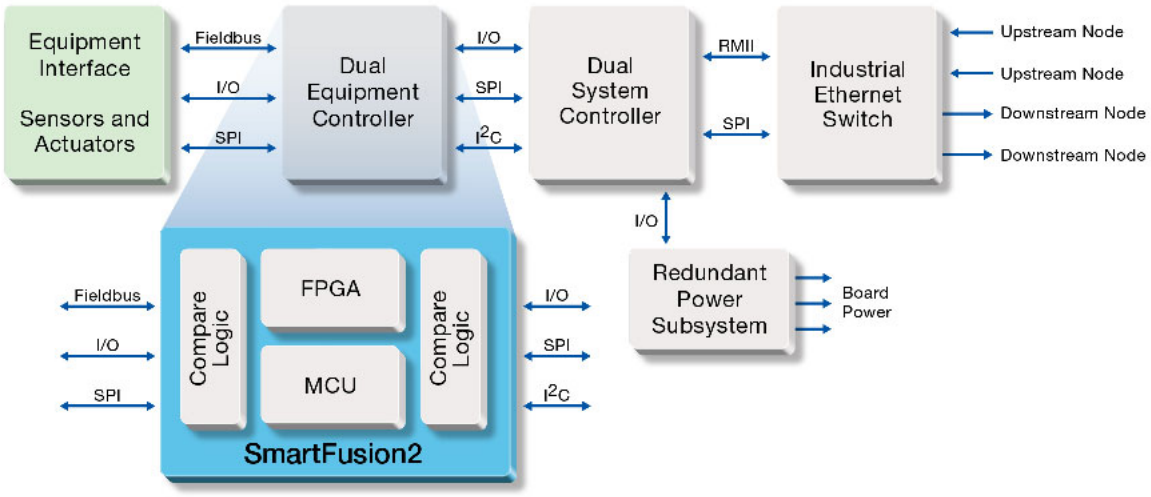

Figure 2: **Example Design with DMR and Design Diversity**

Satisfying Safety Requirements Using Microsemi SmartFusion2 SoC FPGAs

SmartFusion2 SoC FPGAs have been designed from the ground up to implement some of the most stringent safety requirements. As shown in Figure 3 on page 5, the SmartFusion2 FPGA provides a hardened Microcontroller Subsystem with a full MCU along with key peripherals, a hardened Security Controller featuring encryption, error detection, and low-power control, SEU free flash FPGA fabric with SEU protected SRAM blocks, SERDES and a Single Error Correction Double Error Detection (SECDED) external memory interface. Thus, implementing the example design with a SmartFusion2 FPGA gives us some key advantages—the fundamental architectural advantages of SoC FPGAs as well as the unique reliability features native to a flash-based FPGA implementation. Architectural advantages translate into capabilities like improved integration, design diversity, and the ability to include custom reliability features to the system.

The flash memory related features of SmartFusion2 FPGAs automatically protect a design against several of the most common sources of system errors, without requiring any additional design effort. Let's look at these capabilities in more detail.

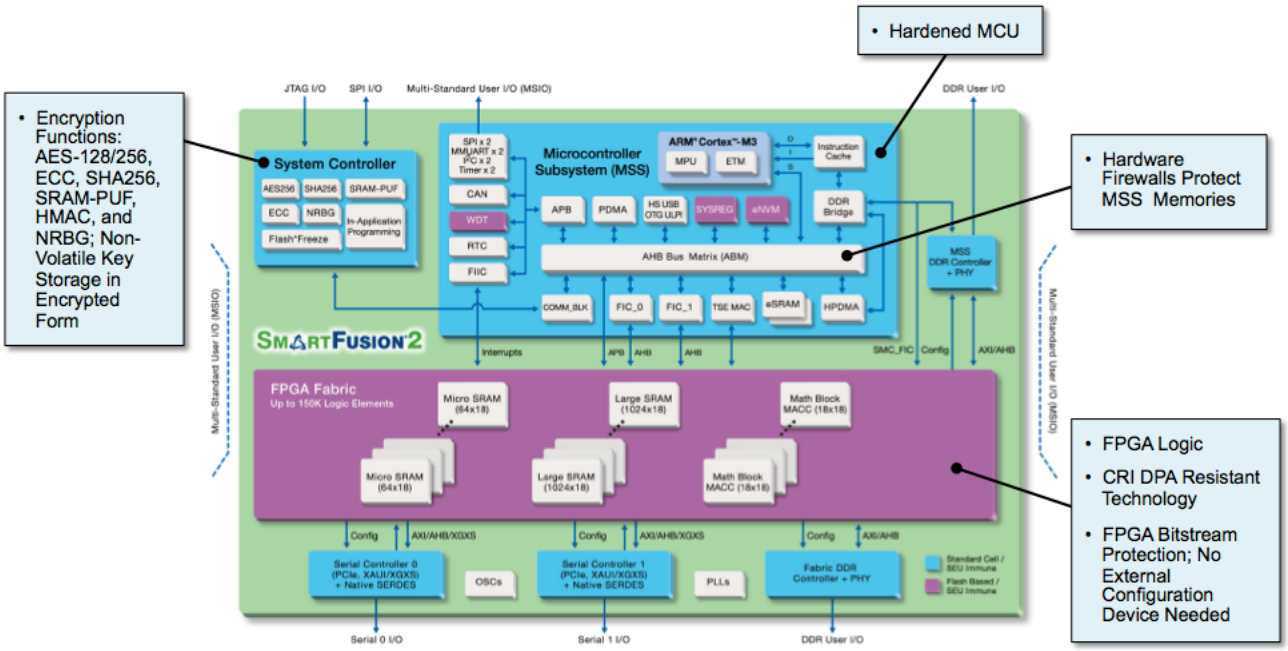

Figure 3: **SmartFusion2 Architectural Block Diagram and Key Features**

Improved Integration

When implementing the example design from Figure 2 on page 4, it would be possible to use separate components for the FPGA implementation of the controller, the MCU implementation and the Compare Logic blocks. These extra components will create new possibilities for errors and system failures, however. An approach that integrates all of these functions into a single device has advantages to the system designer, including better MTBF due to the reduction of components, better cost, and the ability to drive functional safety to smaller systems.

Implementing Design Diversity

The Microsemi SmartFusion2 FPGA architecture, as shown in Figure 3 on page 5, has a hardened ARM®-Cortex™-M3 based Microcontroller Subsystem (MSS) and sufficient FPGA logic to integrate the entire Dual Equipment Controller in a single device, using a design diversity approach while keeping component count the same as our initial, non-redundant, implementation. A SmartFusion2 device would be used for the Dual System Controller, as well as where a single device could integrate the entire function.

Custom Reliability Features

SmartFusion2 devices can implement custom reliability features not found on standard ASSP devices. For example, SmartFusion2 SoC FPGAs have multiple SERDES channels (up to 16) on-chip making it possible to integrate the Ethernet Switch into the Dual System Controller. Once integrated, on-chip additional redundancy, error checking, and advanced error recovery mechanisms could be added in the design. These additional features would create a more reliable design than those available in an ASSP. Other custom features could be added for other communications channels, like Fieldbus or SPI, if an even higher level of reliability was needed.

Note that the inclusion of the SEU protected SRAM blocks and a SECDED external memory controller as features on SmartFusion2 devices simplifies the design of higher level functions, since these memory related reliability features don't need to be designed ‘from scratch’ and won’t require the use of additional FPGA fabric to implement them.

Flash Memory Related Reliability Features

A SmartFusion2 design also benefits from several other ‘built-in’ reliability advantages that come from the underlying flash technology used in their implementation. For example, flash-based configuration memory is immune from SEU events and results in a zero FIT-rate contribution from configuration memory, unlike SRAM-based FPGAs, which are orders of magnitude more susceptible to SEU events. Another reliability feature related to the use of on-chip configuration storage, is that SmartFuson2 devices don't require an external configuration device, unlike SRAM-based FPGAs, which do. The reduced component count for a SmartFusion2 FPGA implementation thus improves system reliability. SmartFusion2 devices also have very low static power consumption, due to the inherent low-power advantages that come from using flash for configuration memory.

Because SmartFusion2 devices use on-chip non-volatile configuration storage, they can support a unique ultra low-power mode, called Flash Freeze mode, that preserves the state of the FPGA configuration and I/Os while dramatically reducing power. The SmartFusion2 device can be put into a low-power ‘sleep’ mode while retaining all the configuration and state information for the design. Upon waking the device, it resumes operation from where it was prior to entering the ‘sleep’ mode. A SRAM-based FPGA can’t do this since they would lose the state of the volatile configuration memory and would require reconfiguration to restore operation. Flash memory typically has much lower static current requirements than SRAM-based FPGA, which resulting in lower overall power requirements. SmartFusion2 FPGAs also benefit from a very low start-up current due to the use of on-chip flash configuration memory, unlike SRAM-based FPGA implementations, which have a large start-up current ‘spike’. These SmartFusion2 FPGA low power features can result in a smaller and less complex power supply design, further improving system reliability.

Security Requirements for the Example Design

Security is an aspect of functional safety that is somewhat new. From a security perspective, functional safety can broadly be defined as being secure in the knowledge that what you designed and built performs the intended functions to an acceptable FIT rate. You must make sure that the components you use to build the system are authentic and that the system is being built as expected.

Lastly, you need to secure the system, when it is in the field, from tampering or intrusion that could introduce errors or system failures. Let's look at each of these requirements in more detail so we can better understand the techniques used to implement a truly secure system.

Securing Your Supply Chain

Without a secure supply chain you could be vulnerable to counterfeiting or fraud. High value devices, like FPGAs and MCUs are particularly vulnerable to counterfeiting where devices are illegally remarked. Often with this for of counterfeiting, the purchased components are completely different from the expected ones— only sharing the same package. The counterfeiter basically pockets the full price the customer pays. With fraudulent devices, perhaps even more insidious than counterfeiting, the thief remarks the correct device, but with a more expensive speed grade or processing level. The thief pockets the difference between the lower grade component and the purchased high-grade device. Unfortunately, the tests typically done at the system level will not show any difference between fraudulent devices and the real thing. Only after being in the field for a while will potential errors (perhaps due to timing violations or lower reliability levels) show up and perhaps lead to functional failures.

Securing Your Manufacturing Process

Once your supply chain is secure, you also need to make sure that your manufacturing process is generating the systems you expect (and only the systems you expect). If systems can be easily copied or cloned, an unscrupulous contract manufacturer could build additional systems and sell them into the distribution channel without your knowledge. If manufacturing tests are bypassed or less qualified components are used, customers will find much lower reliability levels and experience enhanced system failure rates. This can dramatically impact your company brand and your profitability. You need to have a way to secure the manufacturing process even when you use an unsecured manufacturing environments.

Protection from Tampering and Networking Based Attacks

Once you are confident the system you are building is genuine, you need to make sure that your design is protected when it is installed in the field. It must be protected from reverse engineering that could identify potential avenues of attack that an intruder could use to compromise system safety. System tampering, either through hardware probing or network-based remote attacks, need to be protected against. In particular, if the networked system uses a remote field upgrade feature, the data used to make these updates must be protected, secured and authenticated, or an intruder could use this data to insert a Trojan horse to take-over or damage the system. Application data sent to or from the system may also need to be protected if it contains sensitive information related to proprietary processes or information on security keys or protocols.

Satisfying Security Requirements Using Microsemi SmartFusion2 SoC FPGAs

We have identified several key security requirements for our example design. We need to secure the supply chain and manufacturing processes so we produce the systems that we require. We also need to protect our design from invasive tampering or attacks using the remote update capability to make system upgrades and bug fixes over the network. Let's see how we satisfy these requirements using the SmartFusion2 devices.

Protecting Your Supply Chain

To protect the supply chain from counterfeiting and fraud, we need a way to prove that the devices we purchase are what we expect. SmartFusion2 devices have unique features that can aid in verifying that purchased devices are as represented. Each device is shipped with an embedded and digitally signed device certificate (using the x.509 – conforming "Device Certificate") that contains the complete part number, date code, and device version.

The device certificate can be read during manufacturing and compared to the purchase order, verifying that a genuine Microsemi SoC FPGA has been placed on the PCB. This eliminates the possibility of counterfeit or fraudulent devices being used in your system.

You can find out more about these techniques in the articles and videos listed in the "To Learn More" section at the end of this paper under the heading "Supply Chain Assurance".

Protecting Your Manufacturing Processes

Other features of SmartFusion2 devices aid in protecting the manufacturing process. The programming bit stream is encrypted and authenticated so that the design represented by the configuration data isn't available for copying. SRAM-based FPGAs expose the configuration bitstream during programming and are thus very susceptible to copying and cloning. During manufacturing programming of the SmartFusion2 device, a Certificate of Conformance can be generated that cryptographically verifies that the device was programmed with the intended bitstream. This makes it virtually impossible for a contract manufacture to clone or overbuild systems when you use SmartFusion2 devices as the basis for your embedded design.

Protecting Your Embedded System in the Field

SmartFusion2 devices also have several advanced features that make it easy to protect your fielded embedded system from security threats related to reverse engineering, tampering, and network-based attacks using the remote update capability.

Protecting Against Reverse Engineering

SmartFusion2 devices make reverse engineering very difficult, since the FPGA configuration bitstream is stored on-chip. This is much more secure than SRAM-based devices where the bitstream is available for 'snooping' or copying on every system power-up. SmartFusion2 devices also use an encrypted bitstream, so that it is almost impossible to recover the design through bitstream snooping during the infrequent programming operation. Additional protection against reverse engineering is available to eliminate the possibility of an intruder gaining access to the design through JTAG or the debugging interface. These interfaces can be protected by security 'locks' that keep out unauthorized accesses. If desired, locks can be put in place to make the device inaccessible for testing, debugging, or programming. This puts the device into a One Time Programmable (OTP) mode and makes it secure from even the most aggressive intruders.

Protection Against Tampering

One of the most common tampering attacks against secure embedded systems targets the security keys stored within the device. The most invasive techniques 'decap' the device and directly probe the on-chip signals in an attempt to recover security keys or other secret information. SmartFusion2 devices include extensive redundancy and other circuit level techniques to detect and foil these types of attacks.

Another advanced class of attacks uses side-channel information (information that is generated as a side effect of the real world implementation of security algorithms—typically related to signal timing and device power use during security operations) to determine security key values. Differential Power Analysis (DPA), for example, uses statistical measurements of minute changes in device power levels during security operations to identify security key values. Security keys and their related security operations are protected by a variety of circuit design and architecture techniques (used under license from Cryptographic™ Research Inc., a division of Rambus) that defend against side-channel attacks like DPA. (You can read more about these techniques in the articles and videos listed in the "To Learn More" section at the end of this paper under the heading "Protecting Your Design from Side-Channel Attacks").

Using Zeroization as a Tampering Penalty

If tampering is detected, it is important to apply a penalty so that tampering won't obtain any sensitive information. SmartFusion2 devices provide a function to eliminate any sensitive data from the device.

A ‘zeroization’ function can be invoked to erase on-chip secret information with multiple levels of severity. The most severe level can completely erase the device, ‘locking it down’ so that it is impossible to communicate with or use the device at all.

Protecting Your System During Remote Updates

Protecting the system during remote reprogramming updates and upgrades requires a secure communications channel for the FPGA configuration data. SmartFusion2 devices are programmed through a DPA hardened bitstream protocol. Because of this, the built-in security features of the SmartFusion2 device can be used to ensure remotely sourced configuration bitstreams are protected from unauthorized observation and hacking. SmartFusion2 devices can also be programmed through a variety of on-chip interfaces, even through Ethernet, to make remote updates easy to implement. An autonomous programming function is available to manage all the aspects of programming the device with advanced features to recover from power loss during programming. The use of a Known-Good back-up configuration is also supported, so that if somehow the on-chip configuration is incorrect the FPGA can be reloaded with a correct back-up and the system can then execute recovery processes.

A versioning feature is even supported, so that an attack that tries to use an old version of the bitstream (perhaps one with a know exploit) can be detected and thwarted. It’s easy to see that SmartFusion2 devices offer the most robust security available for implementing remote updates for embedded systems.

Conclusion

The advanced safety and security capabilities of Microsemi SmartFusion2 FPGAs satisfy all the key requirements needed for the most robust implementation of a safe and secure embedded process controller. As illustrated in an example design, safety requirements identified by IEC 61508 and the associated Safety Integrity Levels could be addressed by using redundancy (through a design diversity technique) while the fundamental reliability advantages of flash memory (like SEU immune FPGA configuration memory, low power, and improved system integration) provided a solid ‘base’ on which to implement a functionally safe design. Furthermore, the advanced security features of SmartFusion2 insured a secure supply chain (using on-chip digital IDs) and a protected manufacturing environment (through an encrypted and authenticated configuration bitstream). Protection from reverse engineering (through the use of on-chip configuration memory), tampering (including the use of multiple mitigation techniques against side-channel attacks) and attacks using the remote update process (through a secure and autonomous programming facility) are also easily implemented with SmartFusion2 devices. SmartFusion2 devices are clearly your most safe and secure platform on which to design your embedded system.

To Learn More

Supply Chain Assurance

1. Securing Your Supply Chain Life Cycle White Paper
2. Overview of Supply Chain Assurance of Intelligent IC White Papers

Protecting Your Design from Side-Channel Attacks

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

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**.

55900164-0/11.13