Design of a Secure Control Plane Bridge

Application Example

Introduction

In many communications and networking applications it is necessary to bridge high-performance channels with multiple slower peripherals using I^2C, SPI, or General Purpose I/Os. In many applications these bridges are the 'gateway' to the heart of secure communications and storage. Security features must be integrated into the design to protect confidential data and operations. Highly integrated and feature rich SoC FPGAs can dramatically simplify the implementation of these secure bridges when the designer can leverage built-in blocks with an easy to use design flow that guides the designer through the process of defining the target MCU and peripheral set. When the tools generate a correct by construction hardware platform on which application software can be immediately developed, the design is further accelerated.

This application example will identify the requirements for creating a secure control plane bridge and show specific implementation techniques using Microsemi SmartFusion® 2 SoC FPGAs. Some of the key topics covered include: protection of design IP by using secure bit streams; the use of a Root-of-Trust as the foundation of a secure system; advanced SERDES capability that simplifies the implementation of PCIe® bridges; and the advantages of using an advanced tool flow that speeds design implementation.

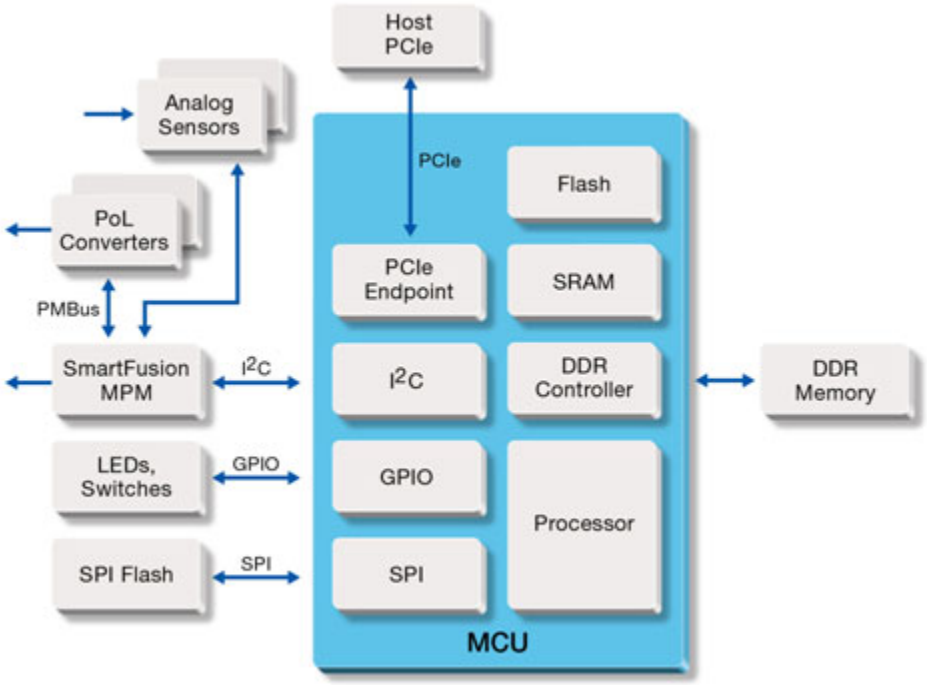

Figure 1: **Block Diagram of a Typical Control Plane Bridge Implementation**

One of the more common bridge applications is on a board within a multi-board chassis. A typical block diagram for such a system is shown in Figure 1 on page 1. Each board has a variety of low speed monitoring and control functions LEDs, switches, power management peripherals, sensors and SPI memory (perhaps used for fault logging) that all need to be managed through a chassis control board. Each board thus needs to bridge the higher speed PCIe connection to the slower speed connection used by the monitor and control peripherals, perhaps using a DDR memory to buffer large data packets. Many times an integrated Micro-Controller Unit (MCU) is used since it combines the processor with many of the key peripherals required for the design. Unfortunately, most MCUs lack many of the features required to create a secure system that protects the control plane bridge from a variety of modern threats. Let's look at these threats and the requirements for a secure control plane bridge in more detail so we can better create a secure implementation using an SoC FPGA like the Microsemi SmartFusion2 FPGA.

Security Requirements for the Control Plane Bridge

Given the dramatic increase in attacks against network connected embedded controllers, as illustrated by the widely publicized Stuxnet virus, and more recent security breaches and 'hacks' (even against home security video cameras, see the "References" section at the end of this paper) security has become a 'must have' feature for many embedded control applications. Control Plane Bridges are the gateway to the heart of many networking and communications systems and need to be protected from the even advanced forms of network-based attacks. The most serious forms of these attacks are directed at the security systems itself and attempt to circumvent the security protocols by discovering the security keys stored within the system. These keys are used to protect any sensitive data and if compromised open the system up to hacking and other intrusions. In some cases the intruder can completely 'take over' the control plane and use it as an attack point for the rest of the system or even the entire plant. Proprietary company data, sensitive customer data and even financial transactions might be compromised. In a worst case scenario equipment or processes could be damaged and even lives put at risk.

Establishing a Root-of-Trust

It will be critical to establish a secure core of the control plane bridge that is impervious to attack. This know secure 'core' is usually referred to as a Root-of-Trust (RoT). As a system element, it supports verification of system, software and data integrity and confidentiality, as well as the extension of trust to internal and external entities. The RoT is the foundation upon which all further security layers are created, and it is essential that its keys remain secret and the process it follows is immutable. In embedded systems, the RoT works in conjunction with other system elements to ensure that all critical system transactions are authenticated (from a known reputable source) and secured (through industry standard encryption and decryption operations). In addition, the RoT implements secure functions for key management that insures attackers don't discover or compromise security key settings or processes.

Protecting the Design Intellectual Property

Creating a RoT isn't possible if the underlying design on which it is based isn't itself secure. Attackers could just reverse engineer a portion of the design and use this knowledge to counteract security functions or exploit weaknesses in the design. In a worst case scenario the attackers could replace the original design with their own design that could capture secret information and transmit it to the attacker without the equipment owner ever being aware of the intrusion. Clearly, it is necessary to create a design that is protected from these types of in order to create a secure RoT. The protection of the design intent, is typically referred to as Design Security and includes a variety of mechanisms used to protect the design Intellectual Property (IP) from reverse engineering or copying as well as counterfeiting or fraud in the supply chain. In this paper we will focus only on protection from reverse engineering, but you can refer to the material listed in the "To Learn More" section at the of this paper to learn more about many more Design Security topics that could be important to your design.

An Overview of Microsemi IGLOO2 FPGAs

Now that we have identified a couple of the most important security requirements for the control plane bridge example design we can begin to look at an implementation using Microsemi IGLOO®2 FPGAs. As shown in Figure 2, IGLOO2 devices combine a variety of fixed function subsystem blocks with programmable FPGA fabric. This combination makes it easy to create custom systems and in particular bridging applications where a variety of communication ports, large on-chip SRAM, and flash memory blocks, and off-chip memory are required.

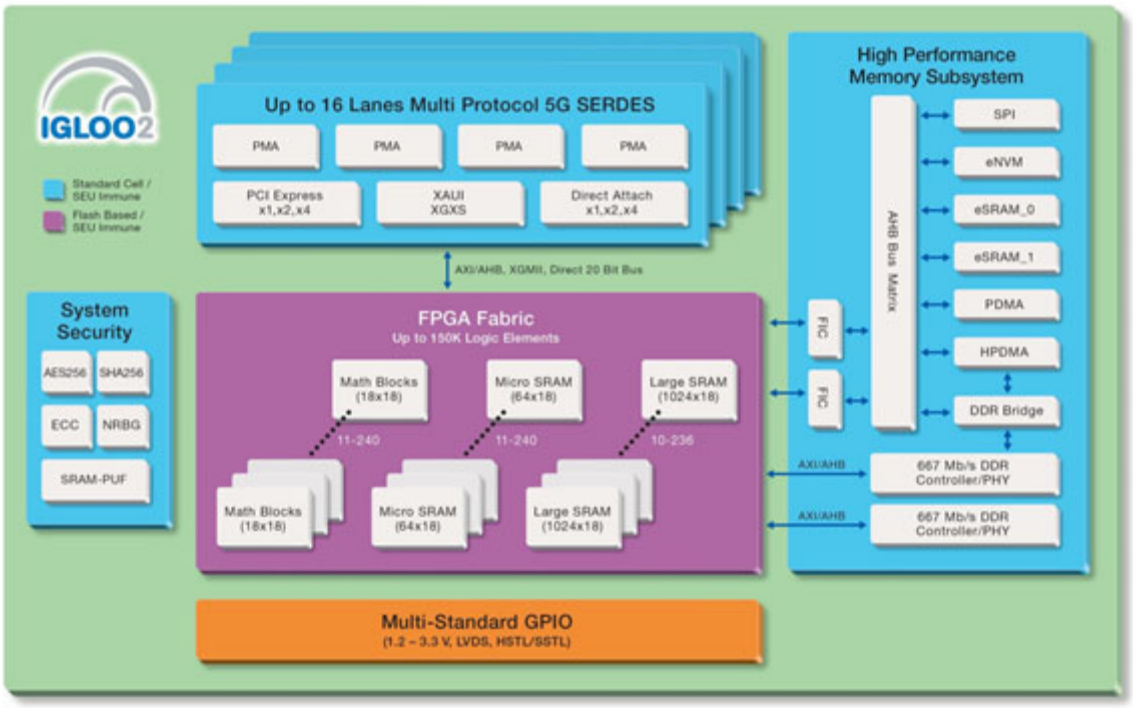

Figure 2: **IGLOO2 Functional Level Block Diagram**

The High-Performance Memory Subsystem (HPMS) includes several advanced features ideal for bridging-large blocks of NVM and SRAM, two Direct Memory Access controllers (Peripheral and High-Performance), a DDR Bridge, two off-chip memory controllers, and an efficient AHB Bus Matrix to connect all the blocks as well as interface them to the FPGA fabric. A System Security subsystem provides important security services and the Multi-protocol 5G SERDES easily implements common standards like PCIe and XAUI/XGXS. We will see how all these key function blocks work together to simplify our implementation of a secure control plane bridge.

Implementing the Control Plane Bridge with Microsemi IGLOO2 FPGAs

The block diagram of a secure control plane bridge implementation using IGLOO2 is shown in Figure 3 on page 4. The High-Performance Memory Subsystem (HPMS) combines the needed on-chip memory oriented blocks as well as DMA, a DDR controller for the external DDR memory, and the SPI port for connecting the external SPI flash device that stores board configuration information and system data used to run the various control functions. The PCIe Endpoint is available as a function within the high-speed SERDES block and connects to the control plane interface. Software response is a critical element in the system.

With up to 32 Interrupts available in the PCIe controller, latency can be reduced since it is easy to determine the appropriate action for any interrupting event.

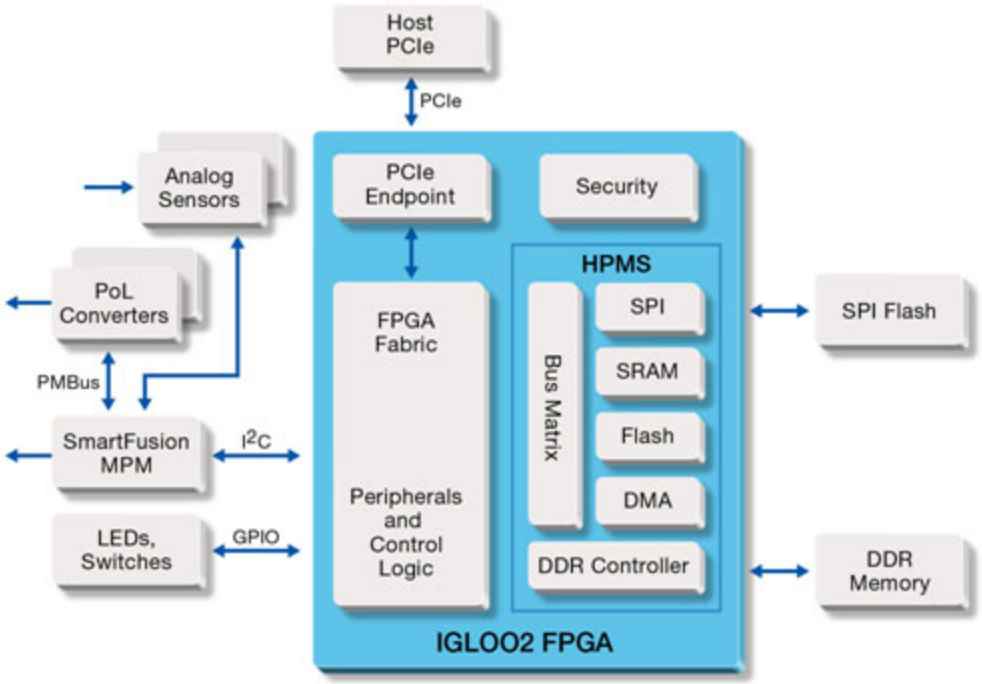

Figure 3: **Secure Control Plane Bridge Implementation Using an IGLOO2 FPGA**

Several low speed peripherals are implemented in the FPGA fabric. The I^2C port is used to communicate with the Mixed-signal Power Manager (MPM) subsystem, based on the Microsemi SmartFusion customizable System-on-Chip (cSoC), that controls the board level Point of Load (PoL) power converters, and connects to the various analog sensors (for temperature, humidity, power rails, and other environmental factors), as well as a variety of digital monitors (fan speed, alerts, data logging, and others). The GPIO port connects to various switches, LEDs and other digital monitoring or control signals. The control for the system is implemented in FPGA fabric and is usually simple enough to require only a modest size state machine. Since the peripherals are implemented in FPGA logic additional intelligence can be built-in, offloading much of the overhead from the controller. This can simplify the controller even more and typically reduces it to just managing interrupts. Setting up and enabling DMA transfers and buffer management functions are usually the most complex tasks.

Satisfying Security Requirements Using Microsemi IGLOO2 FPGAs

Many of the security requirements identified previously are handled automatically by the IGLOO2 FPGA and are related to the underlying implementation of the device. Other features can be overlaid on the fundamental security capabilities of the device to construct higher level security capabilities, like a RoT, that is known to be protected from intrusion and interference.

Protecting Your Design IP from Reverse Engineering

Reverse engineering in particular was identified as a critical concern and IGLOO2 devices have several inherent security features that protect against reverse engineering. IGLOO2 devices make reverse engineering very difficult since the FPGA configuration bit stream is stored on-chip.

This is much more secure than SRAM-based devices where the bit stream is available for ‘snooping’ or copying on every system power-up. IGLOO2 devices also use an encrypted bit stream, so that it is almost impossible to recover the design through bit stream 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.

Security Key Protection

Additional capabilities have been added to IGLOO2 devices to further augment their inherent security characteristics. For example, security keys used for cryptographic operations (like bitstream decryption) are protected by using advanced attack resistant algorithms, and include mitigation techniques for one of the most advanced types of attacks, Differential Power Analysis (DPA). Without the use of these mitigation techniques DPA can be used to recover cryptographic keys used inside a device by observing small statistical (but significant) differences in device power consumption and during cryptographic operations. IGLOO2 devices are the most secure SoC FPGAs available today and are the platform of choice when robust Design Security is required.

Creating a Root-of-Trust

IGLOO2 devices have all the key elements needed to create a robust RoT capability. With a secure starting point, provided by the secure bitstream and protected security key storage and cryptographic processes the higher level functions needed for a RoT can be implemented. For example, cryptographic services (hashing, encryption) can be implemented using simple calls to the IGLOO2 Security Subsystem using a protected on-chip static RSA key stored on the device. In addition, IGLOO2 devices also include features like on-chip oscillators, additional cryptographic services, a true random number generator, and strong design security and anti-tamper features. The advanced computational capabilities of IGLOO2 devices (with FPGA fabric, hardened arithmetic functions, and block memory) along with the breadth of communications capabilities (including many more I/O pins and many built-in high-speed serial interfaces) make for an ideal platform on which to build a robust RoT security system.

Using Data Security Services

Once a robust RoT is established higher level Data Security functions can be implemented without the risk that security keys or algorithms can be compromised. Data Security functions are concerned with the security (confidentiality and authenticity) of data transmitted to/from or stored within the embedded system. The Security Subsystem on the IGLOO2 devices provides a wide range of security services that can be used to authenticate, encrypt, decrypt, and manage security keys. For encryption and decryption the Security Subsystem supplies AES-128 or -256 encryption and decryption functions. SHA-256 is available for computing message digests that can be used for cryptographic functions like secure digital signatures. A Message Authentication Code (MAC) function, HMAC, based on SHA-256 is available to help prove that encrypted data is from the expected sender. KeyTree Key Derivation is also available as an alternative to HMAC. You can refer to the material listed in the "To Learn More" section at the of this paper to learn more about many more Data Security topics that could be important to your design.

Many of these Data Security capabilities can be useful in implementing our control plane bridge. All data traffic can be protected using standard security protocols, (perhaps by using AES-128) so that sensitive data can’t be successfully captured by an eavesdropper. Authentication, making sure messages are from a known and authorized party, can be implemented by using a MAC function (perhaps SHA-256). Advanced challenge-response protocols are also supported to secure the transmission channel between sender and receiver by proving that each ‘knows’ a secret key that a third party can’t identify or impersonate.

Protecting secret data stored in the external SPI Flash memory is also easily accomplished through simple calls to the Security Subsystem. The data in the memory can be authenticated by using one of the MAC functions and then decrypted within the secure IGLOO2 device. Changes to the data can be accomplished by decryption, making modifications, re-encrypting, and then storing data back out to the SPI flash device. 'Plain Text' data is never exposed to the outside world and is one stored within the RoT, making it safe from capture or interference.

IGLOO2 as Your Secure Foundation

These examples show just some of the security requirements IGLOO2 FPGAs can address when implementing embedded systems like a control plane bridge. The security features inherent in the use of flash technology, as well as the variety of additional security features, make it possible to easily protect our control plane bridge from network-based attacks, to store and use confidential data within the system free from eavesdropping and to even protect secret data if the equipment falls into an attackers hands who attempts more invasive hardware based 'hacks'.

Satisfying the Non-Security Design Requirements Using IGLOO2 FPGAs

Now that the security requirements are satisfied we can look in more detail at how to satisfy the other requirements of the design. In particular, we will explore the use of PCIe as the main communications port, how to satisfy the low power operation requirements and how a simplified design flow will allow us to achieve our time to market targets.

PCIe Endpoint Implementation

PCI Express® (PCIe) uses a point-to-point connection architecture as shown on the left hand side of Figure 4 on page 7. PCIe Endpoints are the terminal nodes of the system, similar to the way USB peripherals are located. The PCIe Root Complex is the 'master' of the system and controls the traffic to and from Endpoints. A Switch aggregates multiple PCIe Endpoints to connect to a Root Complex.

In our example design we need to implement a PCIe Endpoint, a common requirement in embedded systems where the system is an end node of the transmission system.

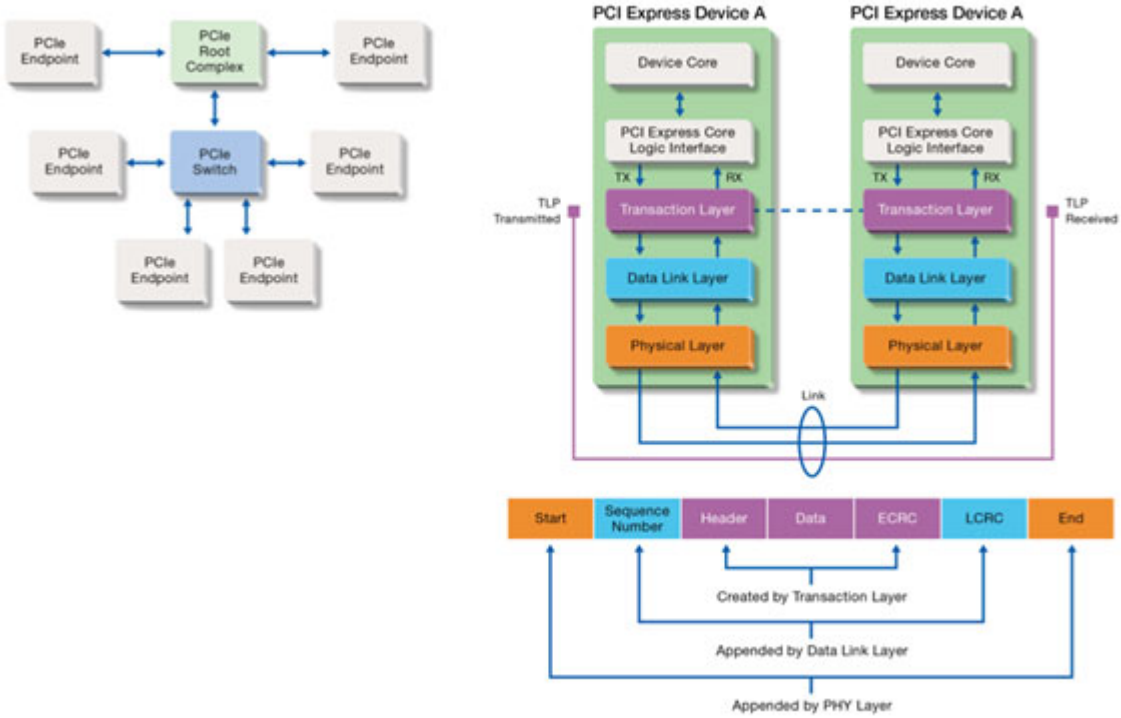

Figure 4: **PCIe Architecture, Protocol, and Transaction Layer Packet**

PCIe uses a familiar multi-layer protocol, show at the top right of Figure 4. The Transaction layer passes user data between PCIe core blocks on linked devices, while the Data Link layer handles flow control and the Physical Layer (PHY) handles link negotiation. Data is encapsulated, as shown in the bottom right of Figure 4, based on the requirements at each layer. In our example design, we need to secure data at the encryption layer by using the encryption and decryption features previously described. The Microsemi PCIe implementation is a 'hardened' block and includes all the key features needed to easily create a standard PCIe Endpoint. Data buffering and management can be handled through the DMA controller within the HPMS using FPGA-fabric memory blocks, or the larger SRAM or NVM blocks within the HPMS. If additional storage is required the external DDR memory can be used, however only encrypted data should be stored in external memory to eliminate the possibility of snooping detecting 'plain text' data that could compromise security. The AHB Bus Matrix makes it easy to manage all these connections and optimize transfer bandwidth.

The IGLOO2 PCIe implementation supports x1, x2, and x4 lane configurations with up to a 2 kbytes maximum payload size to support large block transfers. This can improve efficiency as the overhead for multiple small transmissions is much larger than that for the transmission of a single large block. The variable lane size also makes it possible to 'tune' the lane for the amount of traffic it needs, further improving efficiency. The PCIe Endpoint supports both 64-bit and 32-bit master and slave interfaces to the application layer for additional flexibility.

Low Power Operation

In order to reduce power consumption the PCIe link can be set for an appropriate speed and a minimum number of lanes. Data traffic to the host can be easily consolidated in the SRAM and made available periodically, perhaps every few seconds, to minimize overall bandwidth requirements.

Additionally, the CPB need only initiate a communication to the host when necessary. For example, if a board level alert, like a high temperature or low power warning, has been generated the CPB can initiate a transmission and then wait for the hosts decision on next steps- perhaps a fan speed increase, data logging capture, board reset or even a partial shut-down.

To reduce power even further the SoC FPGA can typically 'turn off' sections of the device not needed while waiting for an alert or a communications request from the host. For example, IGLOO2 devices can put portions of the device in a low power mode, and wake from an interrupt (perhaps due to an external communications or internal timing event), so that dynamic power consumption is significantly reduced. This reduction in dynamic power is in addition to the already low static power consumption of IGLOO2 devices due to the use of flash configuration memory. A Flash*Freeze innovative low power capability, allows IGLOO2 devices to further reduce power consumption while waiting for an alert.

When Flash*Freeze power state is entered, the FPGA fabric is powered down. While in Flash*Freeze power state, the state of the FPGA and related I/Os are maintained, so that upon exit the device continues to operate from where it left off. Flash*Freeze power capability can achieve as low as 1mW standby power while in the low power state.

Simplified FPGA Design Flow

The software flow used to implement the control plane bridge with an IGLOO2 FPGA is illustrated in Figure 5. The design starts with the System Builder, where the designer uses a guided step-by-step process to define the target system including memory, clocking, and peripherals. The resulting correct-by-construction design eliminates the types of errors that crop-up when done 'by hand'.

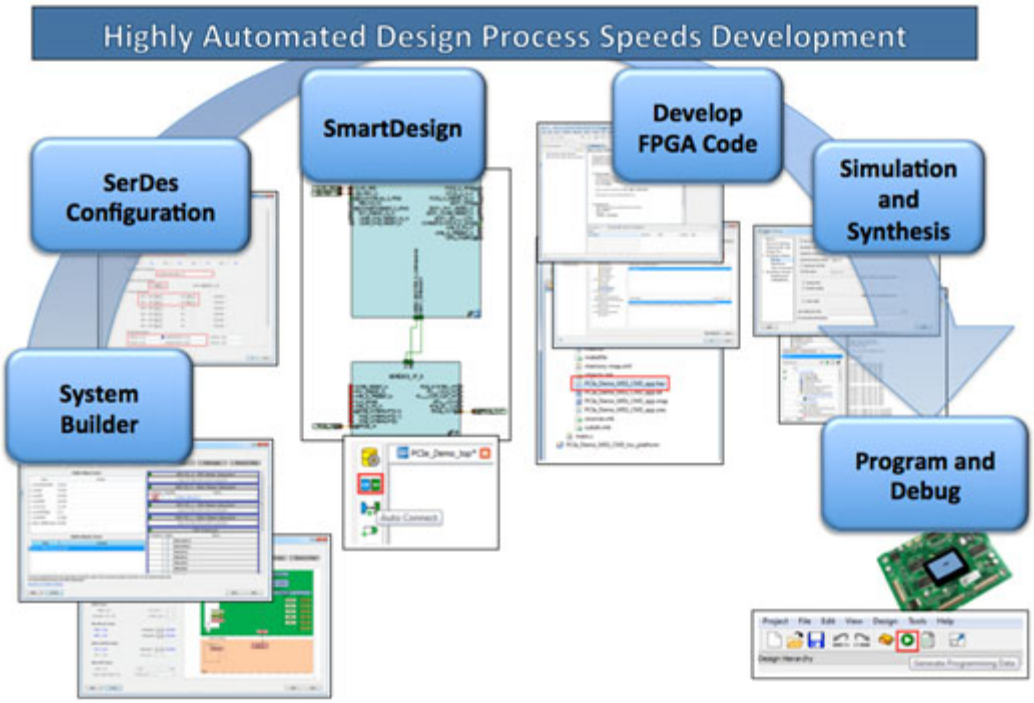

Figure 5: **PCIe Bridge Design Process with IGLOO2 FPGA**

In the next phase, peripheral blocks external to the HPMS, like the PCIe SERDES or fabric-based peripherals are configured through a sequence of GUI-based configuration steps. The process is guided with current selection steps contingent on previous selections, making it easy for the designer to create an optimal target peripheral, free of any configuration conflicts.

The resulting blocks can then be placed into the design using the SmartDesign tool (an Auto-Connect feature) to simplify the interconnection of standard busses.

Standard busses are connected using a single ‘wire’ that automatically connects all the bus signals to the associated sink or source on each block. This significantly speeds block level design and eliminates errors that can result from the tedious process of connecting a large number of individual signals.

Code for the FPGA can be developed using familiar high-level languages and standard Simulation and Synthesis flows. A large number of useful IP Cores, design examples, and reference designs are available to speed FPGA design and development. Once the design has been completely designed and debugged it can be programmed into the target device from within the unified design environment.

Use an Available Evaluation Platform for Development

In many cases using one of the available IGLOO2 evaluation kits from Microsemi can help speed development by leveraging proven and tested code and subsystems. Development of the ‘core’ portion of your design using this approach can help evaluate implementation options and determine your best design approach. In fact, the IGLOO2 FPGA Evaluation Kit, shown in Figure 6, has a variety of key features that make it easy to use as a starting point for a wide range of embedded designs including on-board 512 MB LPDDR memory, 64 MB SPI flash, Ethernet RJ45 connector, and a variety of I^2C, SPI, GPIO, and SMA connectors for SERDES evaluation. The evaluation board is small-form factor PCIe Gen2 x1 lane compliant which allows quick prototyping and evaluation using any desktop PC or laptop with a PCIe slot. Conveniently, a complete PCIe control plane demo is even available and can be used as a starting point for our example design! You can find out more about the IGLOO2 evaluation kit by visiting the webpage shown in the"To Learn More" section at the end of this paper.

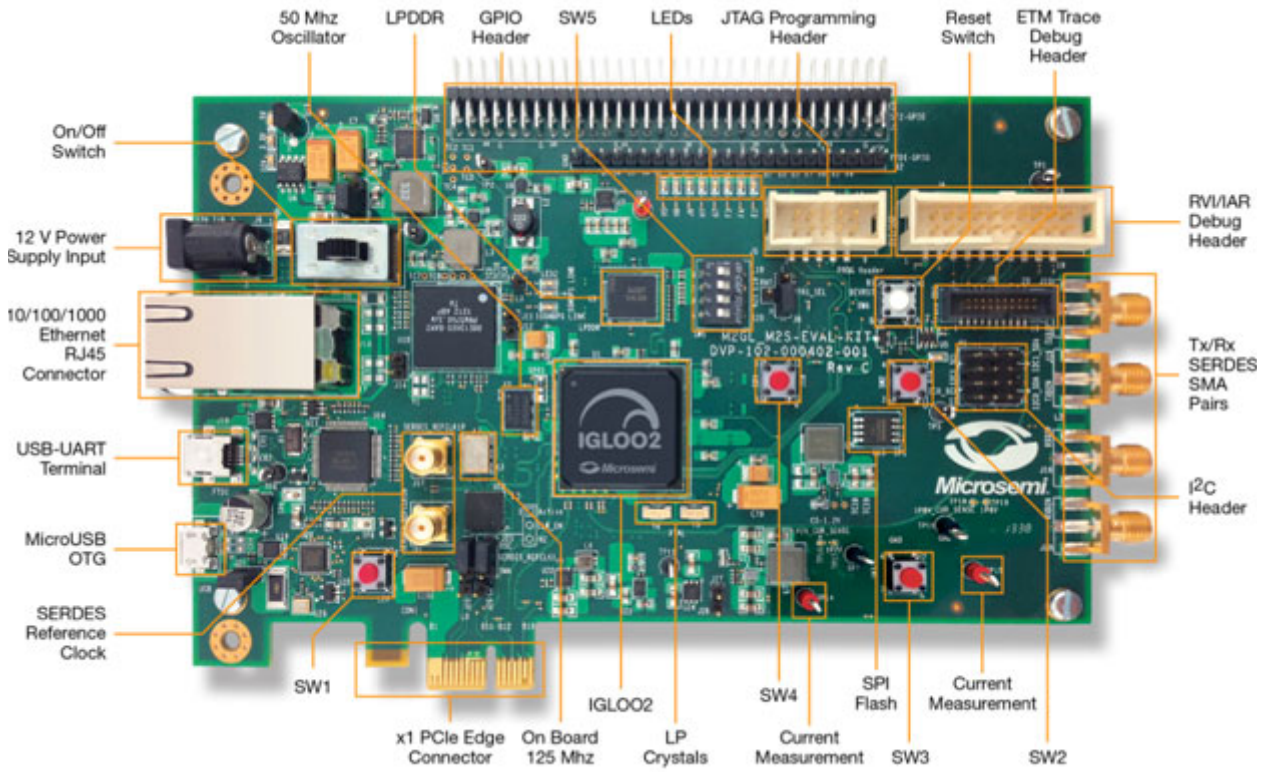

Figure 6: **IGLOO2 Evaluation Board Includes Control Plane Demo Design**

Conclusion

SoC FPGA devices provide all of the key elements needed to implement secure bridging devices, like those used on a Secure Control Plane Bridge. A flash-based SoC FPGA, like Microsemi's SmartFusion2 FPGA, with extensive Design Security features, built-into the device, as well as the advanced Data Security features needed to protect sensitive data is an ideal platform for creating a secure system. When the SoC FPGA has low static power and the ability to dramatically reduce dynamic power (in particular through low power CPU sleep modes and an innovative FPGA fabric sleep mode like Flash*Freeze mode), a low power system is easily implemented. Quick implementation is also possible when a simplified development process that eliminates many tedious and error prone processes is available.

References

"Marketer of Internet-Connected Home Security Video Cameras Settles FTC Charges It Failed to Protect Consumers' Privacy"

To Learn More

Design Security

1. It's Easy to protect Your Embedded System from Theft
2. Introduction to Implementing Design Security with Microsemi Flash FPGAs
3. Securing Your Supply Chain Life Cycle
4. Securing Your Embedded System Life Cycle

Data Security

1. Overview of Data Security Using Microsemi FPGAs and SoC FPGAs
2. SmartFusion2 and IGLOO2 Cryptography Services
3. Overview of Secure Boot with Microsemi SmartFusion2 SoC FPGAs

IGLOO2 Evaluation Kit Webpage

IGLOO2 Evaluation Kit Webpage

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**.

55900165-0/11.13