White Paper

FPGA Design Security Issues: Using the ispXPGA® Family of FPGAs to Achieve High Design Security

December 2003

Lattice Semiconductor Corporation

5555 Northeast Moore Court
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
FAX: (503) 268-8556
www.latticesemi.com

WP1010

Introduction

In today's complex systems, FPGAs are increasingly being used to replace functions traditionally performed by ASICs and even microprocessors. Ten years ago, the FPGA was at the fringe of most designs; today it is often at the heart. With FPGA technology taking gate counts into the millions, a trend accelerated by embedded ASIC-like functionality, the functions performed by the FPGA make an increasingly attractive target for piracy. Many techniques have been developed over the years to steal designs from all types of silicon chips. Special considerations must now be made when thinking about protecting valuable Intellectual Property (IP) implemented within the FPGA.

The most common FPGA technology in use today is SRAM-based, which is fast and re-configurable, but must be re-configured every time the FPGA is powered up. Typically, an external PROM is used to hold the configuration data for the FPGA. The link between the PROM and FPGA represents a significant security risk. The configuration data is exposed and vulnerable to piracy while the device powers up. Using a non-volatile-based FPGA eliminates this security risk. Traditionally, non-volatile FPGAs were based on Antifuse technology that is secure, but very expensive to use due to its one-time programmability and higher manufacturing costs.

Lattice Semiconductor has a superior non-volatile FPGA solution with the ispXPGA family of FPGAs. The ispXPGA and ispXPLD™ families utilize Lattice's ispXP™ technology, which combines reprogrammable non-volatile cells, and SRAM cells on the same chip. The non-volatile memory is used for storing the device configuration securely on the chip. The SRAM memory holds the working configuration after power-up. The technology used in the ispXPGA family of FPGAs provides high security of the configuration pattern while delivering all the benefits of infinitely reconfigurable SRAM memory.

How Does Your Design Get Pirated?

Pirates work in many ways depending on the goal they want to achieve. Sometimes just copying a system can lead to easy money. While other times stealing the IP and mixing it with their own IP to develop a new system can be the reason to crack open a chip. With significant corporate profits at stake, system security has never been more important.

Cloning

When a pirate only wants to mimic the design to reproduce replicas for sale, cloning is the most typical method. Cloning is when an exact copy of the system is made and sold. Most components can be purchased on the open market. The pirate does not care about the exact functions of each chip, just that they can duplicate them with little or no trouble. For the SRAM-based FPGA, all a pirate needs to do is to intercept the configuration bitstream for the FPGA from the boot PROM and then download it on to the duplicate system. Once the configuration of the FPGA is copied, the functionality of the chip (and system) is now available to the pirate without ever having to determine the exact logic implementation in the FPGA. The pirate has no development costs and no design cycle time.

Reverse Engineering

Determining the exact logical functions of a chip would allow a pirate to duplicate a system or to mix and match other functions to implement new systems. Many ways exist to reverse engineer logic designs. Simple designs can be understood by cycling through the inputs and reading the outputs. But with today's large gate counts, this method can be very difficult and time consuming. Unprotected FPGA bitstreams can easily be analyzed to recreate the original design. The resourceful pirate may also "decap" a chip and microprobe its contents. Thermal imaging, focused ion beams and other probing techniques can help the pirate to map out the internal functions of a chip. Some FPGA technologies are harder to probe than others. Careful design by the chip manufacturer can dramatically slow down the reverse engineering process. The more a pirate values a design and its IP, the more effort they will be willing to spend attempting to reverse engineer the design. Careful consideration should be taken to choose a programmable device technology that protects your valuable designs and IP from pirates.

Overbuilding

Considered one of the easiest and most common forms of IP piracy today, overbuilding occurs when a contract manufacturer builds more than the requested quantity of systems, often copying them down to the last detail. Many components come from common sources and can easily be obtained beyond the original order quantity. If FPGAs are used, typically the manufacturer will have the configuration pattern to place in the FPGA, including the “keys” for encrypted FPGA configuration files. There is nothing to prevent an unscrupulous contract manufacturer from simply building extra systems. The extra systems can then be sold on the open market with the profits going directly to the contract manufacturer and not the design owner.

Theft of Services

Electronic devices are used by consumers to access secure transactions and pay-for-services systems. Examples include set top boxes, where cable and satellite TV companies control what the viewer sees and gambling machines, where electronic components determine whom the winners and losers are. If a pirate can bypass, simulate, or otherwise defeat the security features of these electronic devices, theft of service can be done resulting in considerable losses for the system operator.

Hardware Security Risks

The system designer can go to many levels to protect and secure the system they are developing for market. Many considerations must be made in determining what needs to be secured and how. Implementing a little caution, good design practices and selecting the proper components allows a designer to develop a secure system for a reasonable cost. But how secure are the common building blocks of today’s systems? Selecting the right technology and component can mean the difference between a compromised system and a secure one. The following section provides a framework for thinking about security and then reviews the security attributes of some common logic technologies.

Security Level

At what level and cost should security be considered? IBM performed an in depth analysis of security threats to electronic transaction systems and classified the security levels of a design as shown in Table 1 (1, p. 217). This classification of security levels forms a framework within which to think about the security of chips used to implement logic designs.

Security of IP in a programmable device can present many challenges to would be pirates. Evaluation of the potential loss of revenue due to cloning, reverse engineering or just overbuilding a design must be made. The higher the security requirement, the more expensive a device can become, not just in the purchase price but also the total cost-of-ownership.

The IBM paper noted above also proposed that a transaction security system should be designed to the security level of MODH (1, p. 217). Security protection at the moderately high level (MODH) may have a small cost increase in the overall system, but to protect the system at the HIGH level could make costs increase considerably for design and support of a system. A balance can be made between high security and overall production cost. The same security classification fits well with today’s system level components that must be protected and secured to protect from piracy.

Table 1. Security Level Definitions

Security Level	Definition
HIGH	All known attacks have been unsuccessful. Some research by a team of specialists is necessary. Highly specialized equipment is necessary, some of which might have to be designed and built. Total cost of the attack could be one million dollars or more. The success of the attack is uncertain.
MODH	Equipment is available but is expensive to buy and operate. Cost may range from $50,000 to $200,000 or more. Special skills and knowledge are required to utilize the equipment for an attack. More than one operation may be required so that several adversaries with complementary skills would have to work on the attack sequence. The attack could be unsuccessful.
MOD	Special tools and equipment are required, as well as some special skills and knowledge. The tools and equipment may cost from $5000 to $50,000. The attack may become time-consuming but will eventually be successful.
MODL	More expensive tools are required, as well as some specialized knowledge. Tool cost may range from $500 to $5000. The attack may become time-consuming but will eventually be successful.
LOW	Some security features in place. They are relatively easily defeated with common laboratory or shop tools such as pliers, soldering iron, or small microscope.
Zero	No special security features added to the system. Example: a standard PC in a room with free access.

Good FPGA Security Features

There are many different approaches to achieve secure FPGA designs. The following list suggests nine key characteristics of a good, highly secure, FPGA design environment (5, p. 6):

1. The scheme should provide strong protection against both reverse engineering and cloning.
2. No additional components should be required on the customer board, so there is no cost penalty.
3. There should be no effect on the reliability of the user board or need for additional service in the field.
4. The user should not have to maintain a database of encryption keys in order to allow for future changes to the design.
5. There should be no significant complication to the manufacturing flow for products containing the FPGA.
6. No changes should be required to the CAD tools or design flow. In particular, no information, which could compromise the security of the scheme, should be embedded in CAD tools or their supporting files.
7. The scheme should be compatible with standard CMOS processing.
8. The scheme should be based on well-understood and standardized cryptographic algorithms and usage modes to allow easy analysis of threats and should not depend on ‘security through obscurity’.
9. The scheme should be upward compatible with standard programming modes and standard non-volatile memories. It should allow for design upgrades in the field and design changes during prototyping.

Security of ASIC Technology

ASICs (Application Specific Integrated Circuits) are still widely used components in system level designs, even though FPGAs are quickly replacing them in today’s system designs due to their increasing capacity and quicker design cycles. At first glance it may appear that ASICs are a more secure technology than FPGAs. But ASICs are actually one of the easier technologies to reverse engineer.

Many companies exist today that can take an ASIC device and de-layer it, building the schematic of its functions directly from the visible layout of the chip’s individual layers. Although this technique can be costly, if the IP value is

high enough, pirates have been known to reverse engineer chips. An Intel 80386 was reverse engineered in this way in a matter of weeks (2, p. 10). Due to the costs involved in reverse engineering an ASIC, it can be considered to have a moderate (MOD) security level within a system.

SRAM FPGA

The typical FPGA today uses SRAM to hold the functional configuration pattern. When the power is removed, the SRAM loses the stored values. The FPGA must be configured again from a non-volatile memory such as a PROM or Flash memory upon power-up. Because the FPGA and PROM are two separate devices, an external path exists that allows the configuration bits to be transferred from the memory to the FPGA. On power-up or by changing the state of a few pins, the FPGA will be configured automatically by the external memory.

The data link between the FPGA and the PROM represents a significant security risk, as these connections are exposed and easily accessible. When the data is being transferred from the memory to the FPGA, a probe and logic analyzer can easily capture the data stream. Because the bitstream can be snooped so easily between an SRAM FPGA and the Boot PROM, the SRAM FPGA is considered to have a LOW level of security. Most FPGAs currently available from vendors such as Xilinx and Altera fall into this category.

Encryption of SRAM FPGA Configuration pattern

To improve on the security level of the SRAM FPGA, some vendors have begun to implement encryption of the configuration bitstream. Data encryption techniques require that a “key” datastream be embedded with the configuration data stream during the encryption process. This same “key” must be present in the target FPGA in order to decrypt the datastream. In order to keep this key available in the SRAM-based FPGA, the SRAM cells containing the key must remained powered via an on-board battery. Although the encryption scheme is very difficult to crack, other support issues bring into question the overall effectiveness of encryption at the system level. Two issues must be considered before implementing encryption: overall system reliability and security key maintenance.

Batteries have a limited life, especially at high temperature. Thus provision must be made to replace them. The system designer has a tough choice, either they can utilize a mechanical connection or solder the battery to the board. If a mechanical connection is chosen the concern is that if the board were to be bumped in the right way, or be subject to vibrations, the battery could easily lose contact/power and then the decryption key is lost. If the battery is soldered to the board in field replacement becomes challenging at best.

Key management is a factor in the overall security of this kind of FPGA. Not only must the proper key be kept with a design at the factory for fixes later on, but also many other people may require the key including contract manufacturers and field service people. With so many people having the key, encryption of SRAM FPGA devices can be considered to have only a moderate (MOD) security level. Currently this encryption method is implemented on a limited number of Xilinx devices.

One Time Programmable FPGA Technology

One Time Programmable (OTP) technology FPGAs are considered by most to be very secure. For OTP FPGAs, no bitstream needs to be used or maintained for powering up the device because the FPGA configuration is permanently burned on the chip. Today, OTP FPGAs are primarily based on Antifuse technology. Antifuse technology does not lend well to probing by any means. Because of the difficulty scanning antifuse links and the fact that an external boot PROM does not need to be used, the antifuse FPGA has a moderately high (MODH) security level.

Unfortunately this MODH security capability comes at a high price. Antifuse FPGAs are expensive and inflexible. Once the device is programmed, it cannot be reprogrammed again. If changes need to be made to the system later for any reason, the antifuse FPGA device must be discarded and replaced with a new device. Most of the time, this means the whole board must be discarded (or reworked, if practical). OTP technologies also limit production test flexibility. No manufacturing test patterns can be cycled through the part to assist in testing the system, which can add costs due to increased test time.

Non-volatile Reprogrammable FPGA Technology

Non-volatile Reprogrammable FPGAs, based on Flash or EEPROM technology, are considered to have a higher security level than the volatile SRAM-based FPGAs. No configuration bitstream is required at power-up as FPGA configuration is stored in non-volatile memory on the chip. Robust security schemes are available to prevent the readback of configuration data and in today's multi-layer small geometry processes, direct probing of the non-volatile memory cells is very challenging. Because of the difficulty defeating the security scheme, the inability to probe the non-volatile cells and the fact that an external boot PROM is not required, today's non-volatile FPGAs have a moderately high (MODH) security level. Two types of non-volatile memories are being used today; Multi-Voltage and In-System Programmable (ISP™).

Multi-Voltage Non-Volatile FPGA

The Actel ProASIC Plus family of FPGAs represents the industry's only example of multi-voltage non-volatile technology. As noted above these devices provide a moderately high (MODH) security level.

The obvious drawback of multi-voltage non-volatile technology is that it requires two non-standard voltages, in conjunction to its core and I/O voltages in order to reprogram the device. Multi-voltage devices require that a positive 16 volts and a negative 13.5 volts be provided simultaneously along with the core and IO voltages to support device programming. The multiple, non-standard voltages must either be generated on the board or these devices must be configured using a third party programmer prior to placement on the board. Multi-voltage non-volatile FPGAs, while secure, bring their own set of manufacturing challenges associated with device programming.

In-System Programmable Non-Volatile FPGA

The Lattice ispXPGA family of FPGAs represents the industry's only example of In-System Programmable (ISP) non-volatile FPGA technology. As noted above these devices provide a moderately high (MODH) security level.

The key advantage of these devices over MV non-volatile solutions is that they do not require multiple voltages be supplied during programming. ISP FPGAs are programmed using only the core voltage of the device. Again, because of the technology used and the fact that an external boot PROM is not required, the ISP non-volatile FPGA have a moderately high (MODH) security level.

The ispXPGA Security Solution

Lattice Semiconductor offers a unique programmable solution for the security-sensitive system designers: the ispXPGA family of FPGAs. The ispXPGA is based on Lattice Semiconductor's eXpanded In-System Programmability (ispXP) technology. The ispXPGA uses a combination of non-volatile and SRAM technology to deliver a logic solution that provides "instant-on" at power-up, a convenient single chip solution with no external path to connect the non-volatile memory with the functional SRAM, and the capability for infinite reconfiguration (3, p. 2-4).

The ispXP family was designed with security in mind. By its nature, an external bitstream path does not exist. Lattice has included in its device a way to secure or lock the configuration data to prevent readback. In addition, the physics of the small dimensions and multiple layers of metal utilized in the ispXPGA makes it nearly impossible to read the configuration data or defeat the security scheme. Finally, it provides a flexible and secure platform in the design, manufacturing and maintenance aspects of the system.

Its MODH security rating combined with its overall ease-of-use makes the ispXPGA the best choice for security-conscious system designs. In addition, when compared to the good FPGA security features as listed earlier, the ispXPGA meets all of these requirements. Although this paper focuses on secure FPGA applications the same technology is also used in Lattice's ispXPLD family of CPLD devices.

Removal of the External Bitstream Path

Standard SRAM type FPGAs require an external memory of some kind to store the configuration when the FPGA is powered down. In order to provide a highly secure device, the external path to the non-volatile configuration memory must be either eliminated or encrypted. In the ispXPGA, the non-volatile configuration memory has been incorporated into the chip, encapsulating the once vulnerable bitstream path. With this approach no battery is needed to support a decryption key, making the ispXPGA a much more reliable device in the field than SRAM solu-

tions requiring an encryption key. Another benefit to this approach is that it provides a single chip solution simplifying system design.

Locking the Configuration with ispXPGA

The ispXPGA devices provide a security scheme that allows the configuration to be locked within the device. Once set, the device configuration cannot be read from either the non-volatile memory or the SRAM. If desired, the entire device can be erased to remove the lock. All these operations can be set through the ispLEVER design software or the ispVM System programming software (7).

Techniques For Achieving Configuration Security

Lattice Semiconductor has been designing secure products for over 15 years and has applied its knowledge of security to make the ispXPGA family of FPGAs its most secure family to date. The ispXPGA has been designed to withstand both invasive and non-invasive attacks. Many invasive techniques have been used over the years to expose and reverse engineer the contents of programmable devices through de-layering and probing. Non-invasive attacks, such as lowering power levels or shortening erase times in the hopes of exposing the internal pattern have also been taken into account in implementing a secure technology for the ispXPGA. Several design techniques have been used to lock in the pattern and greatly reduce any threat of a pirate breaking into a locked ispXPGA device.

Multiple Hidden Security Bits

Each ispXPGA device has multiple security bits placed throughout the device. The security bits are dispersed throughout the chip and not placed off at one edge where they could easily be identified. Having more than one security bit and having the security bits distributed throughout the silicon die significantly reduce the chance that the security bits can be located and deactivated. In order to protect the security bits from being snooped or probed, all of the security bits have been placed under multiple layers of metal, including GND and V_{CC}.

Security Bits Duplicated in SRAM and Non-volatile Memory

Both the non-volatile and SRAM memory on the ispXPGA devices have their own independent security bits to protect each memory space. The non-volatile memory security bits are placed into the SRAM when the device downloads the configuration data into the SRAM. If the non-volatile memory is secured, the SRAM will always be secured. The only way to clear the security fuses for either memory is to ERASE each memory region independently. If the non-volatile memory is secured, the SRAM side can still be configured and used as a non-secured region with a different pattern. When the device is reconfigured from the secured non-volatile memory, the SRAM will again become secured.

Multiple Paths From Non-volatile to SRAM Memory

A one-to-one relationship exists for each non-volatile and SRAM functional and security bit. The non-volatile cells are downloaded to the SRAM cells at a very fast rate and in a massively parallel fashion. This massively parallel download from non-volatile to SRAM memory makes the pirate's task of trying to locate and then probe each download path virtually impossible. The data path between the standard SRAM FPGA and its memory is now encapsulated and spread out over the entire ispXPGA device.

Trying to Break the Lock

The ispXPGA's silicon design has created a very difficult lock to break from the inside using invasive techniques. Also, it is natural to wonder if non-invasive techniques can be used. Could there be a way to break the lock from the outside, such as partial erasing or partial power down?

It has been suggested that SRAM may be vulnerable if the chip power is lowered to a point that the security bits will flip, but the architecture bits will not. In the ispXPGA, the security bits are placed within clusters of architecture bits. By placing the security bits with the architecture bits, what happens to one happens to the other. Even if a pirate got lucky and was able to clear one security bit without disturbing the architecture bits, the pattern is still protected because all the security bits must be erased to have access to the pattern.

What if the erase instruction is given and then stopped before complete erasure of the device, could the security bits be erased with the architecture bits still intact (programmed)? Again because the ispXPGA technology has

clustered the security bits and architecture bits, as the security bits are getting erased, so are the architecture bits. The only way to deactivate all the security bits on either the SRAM or non-volatile side is to erase the entire memory.

What about a back door through a special manufacturing mode? Raise a voltage here, wiggle a pin there in a special way, and then the device would be available. All it would take would be one pirate to discover the back door and then all the security schemes designed into the ispXPGA family will be for not. It is Lattice Semiconductor's policy that no manufacturing back door be allowed to exist. When a pattern is secured into a device, it is secured. Not even Lattice Semiconductor has access to a secured device. The only way Lattice Semiconductor can access a secured area is to erase the memory, thus clearing the pattern.

ispXPGA Ease of Use

The security features implemented into the ispXPGA provide a robust security scheme with the flexibility and full-features of an in-system programmable device. The SRAM cells in the ispXPGA devices support infinite reconfigurability. The ispXPGA family of FPGAs eliminates all the drawbacks associated with FPGA implementations. The ispXPGA family simply delivers a secure solution without any of the compromises.

Easy To Use In Development

The ispXPGA allows easy system-level development. Reprogramming of the ispXPGA is easy with its ISP technology, supported by both the JTAG and sysCONFIG ports. As the board is being developed and debugged, multiple patterns can be implemented until a final pattern is determined. When configuring the device through the JTAG port, before the device is secured, a Verify is done, providing confidence in the contents of the device. Once the device is secured, an "all zero" pattern will be read back out until the device is erased.

The ispXPGA is an excellent candidate for ease of board development. It eliminates the need for a support memory chip, helping reduce the chip count and area. No extra batteries are needed to support the security features. No extra or non-standard voltage levels are needed to support on-board programming.

Manufacturing Security with the ispXPGA

One advantage of using the ispXPGA in a contract-manufacturing environment is that system security and system testability can both be accommodated. The system designer does not have to worry that a contract manufacture might buy the same chip set and use a pattern supplied by the designer to overbuild systems. The system designer can choose to provide its manufacturing subcontractor with secured, pre-patterned device in the exact quantity required to support the manufacturing build. The manufacturing sub-contractor has no ability to produce more systems than the quantity supported by the pre-patterned inventory. For system-level testing which takes advantage of the ispXPGA ability to reconfigure itself as part of a self-test process flow, the SRAM cells are fully available to the sub-contractor to be used to control the functionality of the device for all the test patterns. Once the assembly and test of a board is complete, the secured non-volatile memory can be refreshed into the SRAM for normal, secured operation. No patterns were ever sent to the manufacturing sub-contractor, insuring that only the number of devices sent would be the number of boards returned.

Conclusions

The Lattice Semiconductor ispXPGA family of FPGA devices provides a moderately high level of security that does not compromise the ease-of-use of the device. The ispXPGA family is an SRAM-based FPGA that eliminates the security risks associated with the exposed external bitstream and need for an external memory for reconfiguration. The technology used to develop the ispXPGA provides excellent protection from both invasive and non-invasive attacks. The ispXPGA family of FPGAs enables highly secure system designs without any of the compromises required by other FPGA approaches.

References

1. Abraham, D.G., G.M. Dolan, G.P. Double, and J.V. Stevens. 1991. Transaction Security Systems. *IBM Systems Journal* vol. 30, no 2. New York: International Business Machines Corporation: 206 – 229.

2. Actel Corporation. Design Security in Nonvolatile Flash and Antifuse FPGAs Security Backgrounder. 2002. Actel Web Site: www.actel.com: p. 1-16.

3. Agrawal, Om. Non-Volatility and Infinite Reconfigurability in PLDs. Lattice Semiconductor Corp. Lattice Web site: www.latticesemi.com: p. 1-4

4. Dipert, Brian. Cunning Circuits Confound Crooks. *EDN Magazine* (October 12, 2000): p. 103-112.

5. Kean, Tom. Secure Configuration of Field Programmable Gate Arrays. Aglotronix Web Site: www.aglotronix.com

6. Kochar, Mehul. Security in QuickLogic Devices. QuickLogic Web Site: www.quicklogic.com

7. Tech Note TN1047, Lattice Semiconductor: December 2003: Lattice Web Site: www.latticesemi.com

8. Telikepalli, Anil. Is Your FPGA Design Secure? Xcell Journal (Fall 2003): Xilinx Web Site: www.xilinx.com

Technical Support Assistance

Hotline: 1-800-LATTICE (North America)
+1-408-826-6002 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com