FPGA Design Security Issues:

Using Lattice FPGAs to Achieve High Design Security

A Lattice Semiconductor White Paper

September 2007

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introduction........ 3
How Does A Design Get Pirated? 3
  Cloning........ 3
  Reverse Engineering 4
  Overbuilding 4
  Theft of Services 4
Hardware Security Risks........ 4
  Security Level 5
  Good FPGA Security Features........ 6
  ASIC Technology 7
  SRAM FPGA........ 7
  One Time Programmable FPGA Technology........ 7
  Non-volatile Reprogrammable FPGA Technology 8
    Hybrid Non-Volatile FPGA........ 8
    Monolithic Non-Volatile FPGA 8
  FPGA Configuration Pattern Encryption........ 8
Lattice FPGA Security Solutions........ 9
LatticeECP2 and Lattice ECP2M S-Series FPGAs 10
LatticeXP and LatticeXP2 Non-Volatile FPGAs 10
  Removal of the External Bitstream Path........ 11
  Locking the Configuration with LatticeXP and LatticeXP2 Devices........ 11
  Techniques For Achieving Configuration Security 11
    Multiple Hidden Security Bits 12
    Security Bits Duplicated in SRAM and Non-volatile Memory........ 12
    Multiple Paths From Non-volatile to SRAM Memory........ 12
    Trying to Break the Lock........ 12
LatticeXP2 Advanced Security Settings........ 13
  Encryption........ 13
  Access Control Settings........ 13
    Configuration Security Bit........ 13
    Flash Protect........ 13
LatticeXP and LatticeXP2 Ease of Use 14
  Easy To Use In Development 14
  Manufacturing Security with the LatticeXP and LatticeXP2 14
Conclusions........ 15
Technical Support Assistance........ 15

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 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 configuration data is exposed and vulnerable to piracy while the device powers up, therefore presenting a significant security risk if the data is not encrypted. Bitstream encryption provides protection as long as the encryption key is unknown, however, non-volatile FPGAs eliminate 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 high manufacturing costs.

The LatticeECP2™ and LatticeECP2M™ S-Series of SRAM FPGAs support encryption of the configuration bitstream to protect against pirating. For even higher security, Lattice Semiconductor has superior non-volatile FPGA solutions with the LatticeXP™ and LatticeXP2™ families of FPGAs. These families utilize Lattice's ispXP™ technology, which combines reprogrammable Flash cells and SRAM cells on the same chip. The embedded Flash 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 Lattice's non-volatile FPGAs provides high configuration pattern security while delivering all the benefits of infinitely reconfigurable SRAM memory.

How Does A 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 for the pirates, while other times, stealing the IP and mixing it with their own IP to develop a new system can be the reason to pirate 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 occurs when an exact copy of the system is made and sold. Most components can be purchased on the open market. The pirates do 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 to the duplicate system. Once the FPGA configuration is copied, the functionality of the chip (and system) is 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 re-create 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 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

Consumers use electronic devices 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 the winners and losers. 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

System designers can go to many levels to protect and secure the system they are developing for market. There are many considerations when 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 – Design Security Levels (1, p. 217). This classification of security levels forms a framework 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 system design and support. A balance can be made between high security and overall production cost. The same security classification applies to today's system level components that must be secured and protected from piracy.

Table 1 – Design Security Levels

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 (8, 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.

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 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 source 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 is easily snooped 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.

Several SRAM FPGAs also support configuration pattern encryption, which is discussed in a separate section of this paper.

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 itself 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 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 technology, are considered to have a higher security level than the volatile SRAM-based FPGAs. No configuration bitstream is required at power-up because the 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 complexity of 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.

Hybrid Non-Volatile FPGA

The Xilinx Spartan-3AN family of FPGAs is an example of a hybrid non-volatile solution. The name hybrid refers to the fact that the FPGA consists of a standard SRAM FPGA die plus a separate SPI Flash memory die, both in one package. In the stacked die FPGA the configuration bitstream travels from the SPI Flash memory to the SRAM FPGA, leaving it vulnerable to bitstream snooping.

Monolithic Non-Volatile FPGA

The LatticeXP family of FPGAs is the industry's first available example of In-System Programmable (ISP™) non-volatile FPGA technology. The second generation LatticeXP2 family of FPGAs is the industry's first line of true monolithic, 90nm, non-volatile FPGAs. As noted above, both of these families provide a moderately high (MODH) security level.

The LatticeXP and LatticeXP2 single die solutions eliminate vulnerability due to bitstream snooping. Again, because of the technology used and the fact that an external boot PROM is not required, the ISP non-volatile FPGA has a moderately high (MODH) security level.

FPGA Configuration Pattern Encryption

To improve the security level of the FPGA, several vendors implement encryption of the configuration bitstream. Data encryption techniques require a key to transform the configuration datastream into a different datastream of the same size during the encryption process. This same key must be present in the target FPGA in order to decrypt the datastream. A strong encryption algorithm must meet the following criteria (10, p. 2):

- There must be no way to find the unencrypted clear text if the key is unknown, except by brute force, i.e. to try all possible keys until the right one is found.
- The number of possible keys must be so large that it is computationally infeasible to actually stage a successful brute force attack in a reasonable length of time.

In 2002 the Advanced Encryption Standard (AES) was approved by the National Institute of Standards and Technology for federal use in the United States. The AES algorithm can be implemented with 128, 192 or 256 bit keys (3, p. 5).

In order to keep this key available in the SRAM-based FPGA, the SRAM cells containing the key must remain powered via an on-board battery, or the key must be stored in the FPGA using antifuse technology. Batteries have a limited life, especially at high temperature, so provision must be made to replace them. System designers have 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 vibration, 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. Currently this encryption method is implemented on a limited number of Xilinx devices.

The LatticeECP2/M S-Series FPGAs utilize fused technology to store the 128-bit AES key. Fused technology protects the key within the FPGA and eliminates the reliability concerns with using batteries.

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 other people also 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.

Lattice FPGA Security Solutions

All Lattice FPGAs provide configuration data read security, meaning that a fuse can be set so that when the device is read all zeros will be output instead of the actual configuration data. This kind of protection is common in the industry and provides very good security when the configuration data storage is on-chip, such as with the LatticeXP, LatticeXP2 and MachXO™ device families. However, if the configuration bitstream comes from an external boot device it is quite easy to read the configuration data, allowing access to the FPGA design. For this reason, the LatticeECP2/M S-Series FPGAs support 128-bit AES. The security features of Lattice’s FPGAs are highlighted in Table 2 - Lattice FPGA Security Features.

Table 2 - Lattice FPGA Security Features

Family	128-Bit AES Support	On-Chip Flash with Security Settings	Flash Protect	OTP Mode
LatticeXP2	✓	✓	✓	✓
LatticeXP		✓		
LatticeECP2/M S-Series	✓			

LatticeECP2 and Lattice ECP2M S-Series FPGAs

The LatticeECP2/M S-Series supports 128-bit AES to protect the configuration bitstream. The user selects and has total control over the 128-bit key. The encryption key is programmed into the OTP fuses in the FPGA. This step is separated from file encryption to allow flexibility in the manufacturing flow. For instance, the board manufacturer might program the encrypted file into the SPI Serial Flash, but the key might be programmed at the user's facility. This flow adds to design security and it allows the user to control over-building of a design. If the key is programmed at the factory, then the factory controls the number of working boards that enter the market. The LatticeECP2/M S-Series will only configure from a file that has been encrypted with the same 128-bit key that is programmed into the FPGA.

LatticeXP and LatticeXP2 Non-Volatile FPGAs

Lattice Semiconductor offers unique programmable solutions for security-sensitive system designers: the LatticeXP and LatticeXP2 families of non-volatile FPGAs. These families are based on Lattice Semiconductor's eXpanded In-System Programmability (ispXP) technology. The LatticeXP uses a combination of embedded Flash 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 (4, p. 2-4). The LatticeXP2 family combines the benefits of the LatticeXP family along with a new architecture referred to as flexiFLASH™ to achieve enhanced functionality and security (6, p. 8-11).

Both the LatticeXP and LatticeXP2 families were designed with security in mind. By their nature, non-volatile devices have no external bitstream path. 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

LatticeXP and LatticeXP2 make it nearly impossible to read the configuration data or defeat the security scheme. Finally, these families provide a flexible and secure platform in the design, manufacturing and maintenance aspects of the system.

Their MODH security rating combined with their overall ease-of-use make the LatticeXP and Lattice XP2 devices the best choices for security-conscious system designs. In addition, when compared to the good FPGA security features listed earlier, the LatticeXP and LatticeXP2 FPGAs meet all of these requirements. This section will first discuss the security features designed into both the LatticeXP2 and LatticeXP families, and then it will discuss the advanced security features available in the LatticeXP2 family.

Removal of the External Bitstream Path

Standard SRAM FPGAs require an external memory 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 LatticeXP and LatticeXP2 devices, the non-volatile configuration memory is incorporated into the FPGA die, resulting in a single chip and encapsulating the once vulnerable bitstream path. This single chip solution enhances security and simplifies designs.

Locking the Configuration with LatticeXP and LatticeXP2 Devices

The LatticeXP devices provide a security scheme that allows the configuration to be locked within the device. Once set, the device configuration cannot be read from 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.

The configuration can also be locked within the LatticeXP2 FPGA; however, the LatticeXP2 device takes this one step further by preventing erasure.

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 LatticeXP and LatticeXP2 families of FPGAs secure from piracy. Both families were designed to withstand 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 LatticeXP and LatticeXP2 devices. Several design techniques have been used to lock in the pattern and greatly reduce any threat of a pirate breaking into a locked LatticeXP or LatticeXP2 device.

Multiple Hidden Security Bits

Each LatticeXP and LatticeXP2 device has security bits for SRAM and Flash. The security bits are placed in the “center” of the design to avoid easy identification. 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 Flash and SRAM memory on the LatticeXP and LatticeXP2 devices have their own independent security. The Flash memory security setting is placed into the SRAM when the device downloads the configuration data into the SRAM. If the Flash 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 be secure.

Multiple Paths From Non-volatile to SRAM Memory

A one-to-one relationship exists between Flash and SRAM functional and security bits. 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 encapsulated and spread out over the entire LatticeXP or LatticeXP2 device.

Trying to Break the Lock

The LatticeXP and LatticeXP2’s silicon design creates 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 is reset, but the configuration is not. In the LatticeXP and LatticeXP2 devices, bits are clustered to ensure that such an action will not reset the security while a valid configuration remains.

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 LatticeXP and LatticeXP2 technologies cluster the security and architecture bits, both are erased together.

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 LatticeXP and LatticeXP2 families will be for naught. It is Lattice

Semiconductor's policy that no manufacturing back door is allowed to exist. When a pattern is secured into a device, it is secured. Not even Lattice Semiconductor has access to a secured device.

LatticeXP2 Advanced Security Settings

The LatticeXP2 flexiFLASH architecture provides users with advanced security settings reaching beyond the standard levels of security inherent to non-volatile FPGAs. To enhance security further, the LatticeXP2 provides 128-bit AES support and access control settings for multiple configuration security levels. This section steps through the advanced security settings available in the LatticeXP2 family. All these operations can be set through the ispLEVER design software or the ispVM System programming software.

Encryption

The LatticeXP2 family supports the 128-bit AES discussed earlier. The LatticeXP2 family has a hardwired AES decryption engine embedded within each device, which supports secure configuration of the on-chip Flash from an encrypted bitstream. The 128-bit key is stored within the device, and once the key lock is programmed and the power is cycled the key cannot be read back from the device.

Access Control Settings

The LatticeXP2 family offers two access control settings that, when used together, provide several distinct levels of security. The levels range from no additional security to making it a true OTP device. Flexible access control settings allow users to match the level of security to their specific use conditions.

Configuration Security Bit

There is a configuration security bit to control read access of both the Flash and SRAM. Setting the configuration security bit prevents reading of the device. The configuration security bit can be used in conjunction with the Flash Protect settings described below to prevent or allow read access in each scenario.

Flash Protect

There are three Flash Protect settings to control erase and write access to the Flash. The three settings are Off, Lock and OTP. When set to off, there are no restrictions on erasing or writing.

For higher security there is a lock setting that utilizes a 64-bit key to prevent unauthorized erasing and programming. In lock mode, the 64-bit key must be programmed into the device first. If it matches the key stored in the device, then it will enter the programming mode for erasing and programming the Flash and SRAM fuses. If the key is unknown, then a programmed device behaves like an OTP device.

LatticeXP2 devices include a Flash Protect setting to make the device truly OTP, also known as Permanent Lock. The purpose of the OTP capability is to provide the highest level of security for the LatticeXP2 family. Once the OTP setting is programmed it creates a permanent barrier, preventing any access to the contents of the device. When used in conjunction with the configuration security bit the device cannot be erased, programmed or read.

LatticeXP and LatticeXP2 Ease of Use

The security features implemented in the LatticeXP and LatticeXP2 devices provide a robust security scheme with the flexibility and full-features of in-system programmable devices. The SRAM cells in the LatticeXP and LatticeXP2 devices support infinite reconfigurability. The LatticeXP and LatticeXP2 families of FPGAs eliminate the drawbacks associated with FPGA implementations and deliver a secure solution without compromises.

Easy To Use In Development

The LatticeXP and LatticeXP2 devices allow easy system-level development. Reprogramming the devices is easy accomplished through either the JTAG or 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 LatticeXP and LatticeXP2 are excellent candidates for ease of board development. They eliminate the need for a support memory chip, helping reduce the chip count and area. No extra batteries are needed to support the security features, and no extra or non-standard voltage levels are needed to support on-board programming.

Manufacturing Security with the LatticeXP and LatticeXP2

One advantage of using a LatticeXP or LatticeXP2 device 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 manufacturer 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 devices 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 LatticeXP and LatticeXP2 device’s ability to reconfigure 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. In addition, with 128-bit AES support the LatticeXP2 family can provide an even higher level of protection against overbuilding.

Conclusions

Lattice Semiconductor's portfolio of FPGAs offers a broad range of features for security conscious designers. The LatticeECP2/M S-Series provides a moderate security level with 128-bit AES. The LatticeXP and LatticeXP2 families of FPGA devices provide a moderately high level of security that does not compromise the ease-of-use of the device. The LatticeXP and LatticeXP2 families are SRAM-based FPGAs that eliminate the security risks associated with exposed external bitstreams and external memory for reconfiguration. The technology used to develop the LatticeXP and LatticeXP2 devices provides excellent protection from both invasive and non-invasive attacks. The LatticeECP2/M S-Series, LatticeXP and LatticeXP2 families of FPGAs enable highly secure system designs without any compromises.

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. *Federal Information Processing Standards Publication 197*. 2001. Announcing the Advanced Encryption Standard (AES).
4. Lattice Semiconductor. 2005. LatticeXP – Combining Low-Cost & Non-Volatility to Deliver No Compromise FPGAs. Lattice Web site: www.latticesemi.com.
5. Lattice Semiconductor. 2007. LatticeXP2 Configuration Encryption and Security Usage Guide. Lattice Web site: www.latticesemi.com.
6. Lattice Semiconductor. 2007. Third Generation Non-Volatile FPGAs Enable System on Chip Functionality. Lattice Web Site: www.latticesemi.com.
7. Dipert, Brian. Cunning Circuits Confound Crooks. *EDN Magazine* (October 12, 2000): p. 103-112.
8. Kean, Tom. Secure Configuration of Field Programmable Gate Arrays. Aglotronix Web Site: www.aglotronix.com
9. Kochar, Mehul. Security in QuickLogic Devices. QuickLogic Web Site: www.quicklogic.com
10. Seleborg, Svante. 2004. About AES – Advanced Encryption Standard. Axantum Web Site: www.axantum.com.
11. 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)
          +1-503-268-8001 (Outside North America)
e-mail: techsupport@latticesemi.com
Internet: www.latticesemi.com