Actel

Security FAQ

8/15/2002

™

Secured by Actel

This FAQ is intended to serve as a resource to anyone interested in the general fields of information security, semiconductor security, and system security. Much of what this document covers is specific to the concept of FPGA design security and embedded system design concerns for industries that use FPGAs. This FAQ is provided by Actel Corporation and includes commonly asked questions from the embedded design community.

What is "security" in the semiconductor market?

Generally there are two distinct classes of design security. The term is most often applied to the following areas:
a) Design security (also called IP security). In this case the designer wants to prevent alteration or theft of his design. A design may include licensed 3rd party IP as well as internally developed IP. The chief concern is that a competitor will clone or reverse engineer critical designs.
b) Data security. In this case, the designer wants to insure that data being sent to or from the FPGA cannot be copied, altered, or corrupted. This usually involves various encryption schemes and is a very complex topic.

Is there an increased need for security?

Yes, as FPGAs capture more and more of the traditional ASIC market the need for security increases dramatically. A few years ago, FPGAs were viewed primarily as glue logic. FPGAs were relatively small devices that were often used to interface between ASSPs or custom ASICs. Today, FPGAs have achieved new levels of density, a broad range of embedded features, and higher performance than ever before. As a result many designers now use FPGAs as an alternative to ASICs (over 20% of 'ASIC' design starts in 2002). Many systems today have most, if not all, of the sensitive IP contained in an FPGA. A typical system might incorporate a processor/DSP, some memory, a few ASSPs, and one or more FPGAs. If you can read the contents of the FPGA you can duplicate (or enhance) the function of the entire system since all other components are off-the-shelf. The vulnerability of FPGAs to copying puts the intellectual property of the system at risk. Given the continued rapid adoption of FPGAs, security is a growing problem and one that was not as significant a concern when ASICs (which are more secure than SRAM FPGAs) were at the heart of most embedded systems.

How capable are malicious attackers today?

IBM initially proposed a now widely accepted definition of integrated circuit security in the IBM systems journal Vol. 30 No 2. The preliminary studies from IBM grouped adversaries and attacks into three classes, in ascending order, depending on their expected abilities and attack strengths.

Class I: (clever outsiders) A knowledgeable individual using low cost easily available tools or services could reverse engineer a design in a short period of time. These low security solutions are usually included in products such as phone cards, debit cards, and set-top boxes.

Class II: (knowledgeable insiders) A skilled individual or team with access to expensive sophisticated equipment could reverse engineer a design given sufficient time. Individuals involved in reverse engineering at this level are usually associated with a commercial enterprise such as a game copier or other cloned consumer electronics.

Class III: (funded organizations) A highly skilled team, using equipment not commonly available in the commercial market and substantial time could reverse engineer a design. This third category is largely limited to government agencies like the NSA with nearly unlimited funds available for their activities.

It should be noted that security is a relative concept; virtually any protection method can eventually be cracked given enough time and resources. At some point the value of the IP contained within a circuit is less valuable than the effort required to copy it.

Conventional SRAM based FPGAs are susceptible to class I attacks. They can be easily cloned and have been reversed engineered with both activities well documented in various FPGA news groups. For comparison, experts believe all members of Actel's ProASIC and metal-to-metal antifuse families would not be susceptible to anything less than a class II+ attack.

What are the accepted levels of security as viewed today?

In the IBM systems Journal Vol 30, No 2, Abraham ET AL., outline a practical system to define the various physical security levels for modern electronic systems. The attack study initiated by IBM helped create security levels that are related to the strength of an attack. Again, largely accepted as the norm, and very similar to the modern day NSA guidelines, these security ratings help determine the physical security needed to minimize threat from any given attack.

Security Level	Definition
ZERO	No special security features added to the system. Example: a standard PC in a room with free access.
LOW	Some security features in place. They are relatively easily defeated with common laboratory or shop tools such as pliers, soldering iron, small microscope or cheap programmer. A boot prom for an SRAM FPGA is easily copied using these tools.
MODLOW	More expensive tools are required, as well as some specialized knowledge. A security scheme using a couple of security bits may be an example here.
MOD	Special tools and equipment are required, as well as some special skills and knowledge. The attack may become time-consuming but will eventually be successful. Reverse engineering an ASIC using photographic techniques, for example.
MODHIGH	Equipment is available but is expensive to buy and operate. 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. NVM FPGAs such as ProASIC$^{\underline{\text{PLUS}}}$, for example.
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. The success of the attack is uncertain. Antifuse FPGAs, for example.

What's the difference between "reverse engineering," "cloning," and "over building?"

In reverse engineering, a competitor copies a design by essentially reconstructing a "schematic" level representation from the physical device level. In the process, he understands how your design works, how to improve it and how to disguise it so that it no longer looks like your original work.

Reverse engineering is common with gate arrays and is on the rise with FPGAs. Cloning requires something that is easily copied; this is the situation with SRAM-based FPGA designs. A competitor either makes a copy of the boot prom or intercepts the bit stream from the on-board processor and copies the code. He does not know how your design works, nor does he need to. He is able to steal the entire design merely by copying the external bitstream, which is always required for an SRAM FPGA.

In over-building an unscrupulous manufacturer builds more systems than they were contracted to. The additional parts for the unauthorized boards are sourced on the open market, as most system components are standard products. The over-build inventory is sold by the manufacturer in the open market, often boxed exactly as the legal product. The manufacturer takes the profit with no design, development, or support overhead, essentially skimming profits from the original customer. This is very difficult to detect and is believed to be one of the more common forms of IP/design theft.

Aren't most FPGA products secure?

No, the industry offers a wide range of security coverage depending upon the solution you select. Solutions range from very secure nonvolatile Flash and antifuse architectures to non-secure SRAM products.

Industry experts typically agree antifuse is the most secure architecture available.
A number of factors complicate attempts to compromise an antifuse FPGA. In order to determine the state of any given fuse, the microscopic size and sheer number of the antifuses make it essentially impossible to locate each fuse and identify its programming state. For example Actel's new 2 million gate AX2000 antifuse FPGA contains approximately 53 million antifuses with only 2-5% programmed in an average design. Invasive probing to evaluate each fuse would most likely result in the destruction of the very programmed states needed to trace the design.

Certain types of Flash technologies are also extremely secure, and definitely more secure than SRAM based FPGAs. Once a Flash FPGA is programmed the configuration contents do not need to be reloaded each time power is applied to the system. As no physical change actually occurs at the silicon level, it is also impossible to determine the state of a device through intrusive probing surveys. Some suppliers have also taken steps to employ additional protection schemes involving an access key. Actel's new ProASIC$^{\underline{PLUS}}$ family uses keys that range from 79 bits to 263 bits in length. Once the key is used to secure the Actel ProASIC$^{\underline{PLUS}}$ device the contents cannot be read-out without first unlocking the device. Furthermore, the time taken to exhaustively test all key combinations would run in to the billions of years.

SRAM-based products are the least secure of all technologies. Since SRAM-based FPGAs are volatile, they must be “initialized” (or configured) at power on. The bitstream used to initialize an SRAM FPGA is typically loaded from an on-board configuration device. This bitstream can be intercepted in route at the circuit level and replicated. This configuration data can be read from the configuration device and manipulated or copied, or the on board PROM can be replicated. There have been some solutions advanced, utilizing preconfigured FPGAs with a battery back-up, but this approach requires additional board space for batteries and PROM memories, power, reduced reliability (battery life), and can add significantly to overall system complexity and cost.

Are ASICs a secure technology?

Many people think of ASICs as being a very secure technology, especially when compared to SRAM FPGAs. In reality, ASICs in general and gate arrays in particular are relatively easy to reverse engineer. A device is decapped and then stripped layer by layer and each layer is photographed. The resulting database is a complete layout of the chip and how it functions. This can be a somewhat costly process but several labs throughout the world offer this service publicly. Security is relative; it all depends on how badly someone wants the information contained within the device.

What is DES?

DES is the Data Encryption Standard and uses a 56-bit private key encryption algorithm. It was developed by IBM and the US government and first adopted in 1977. DES is the Federal Information Processing Standard 46-3, which describes the data encryption algorithm (DEA) this is also defined in the ANSI standard X9.32.
The company RSA security has sponsored a series of contests to crack DES. At the DES III contest in January 1999, DES was cracked in 22 hours and 15 minutes. The US Government has gone through a formal process to replace DES. This effort, called the Advanced Encryption Standard (AES) is now increasingly being used in place of DES. Critics have long attacked DES for being inadequate to provide sufficiently high levels of security. DES is fundamentally inadequate because its 56-bit key is too short and is vulnerable to brute-force search. A group of cryptographers looked at key lengths in a 1996 white paper. They suggested a minimum of 75 bits to consider a cipher secure.

What is Triple DES?

The financial services industry developed ANSI X9.52, a standard for so-called 'triple-DES' encryption. In triple-DES, each 64-bit block of a message is encrypted with three successive DES operations rather than one, and the operations involve two or three different keys. Triple-DES offers an effective key size of 112 bits in typical applications, as opposed to only 56 bits for DES. Since triple-DES uses the same basic block size as DES, it is also vulnerable to similar brute force attacks. Triple DES is considered by many to be no more than an interim solution until the widespread adoption of AES. The reality is the actual strength of triple DES is not known at this time but experts believe that it remains vulnerable, hence the push by security experts for the development of AES to supercede DES.

What is AES?

The Advanced Encryption Standard was selected by the National Institute of Standards and Technology in 2000 after a three-year public search to replace DES. From a large field narrowed to just five finalists a solution called Rijndael was voted the best. It is fast and compact with a simple mathematical structure that eliminated much of the cumbersome overhead of 3DES. AES is also easily implemented in hardware, can be integrated into a wide range of electronic systems requiring secure data transactions.

While there are other encryption standards currently in use today, like MARS, RC6, Serpent, and Twofish for example, and in some instances they may even be more efficient than AES for certain applications, but AES is already being used in next generation projects and retrofits to existing or legacy encrypted Enterprise solutions because it has the assurance of a federally endorsed national standard.

What are FPGA vendors doing to improve security?

Most FPGA vendors have done little or nothing to improve device security. If security is a factor in your selection process, focus on products that utilize inherently secure technologies like antifuse and certain types of Flash. One SRAM vendor has instituted a security scheme that involves battery back up of pre-configured devices, though many regard this solution as expensive, cumbersome and difficult to implement.

What kind of techniques can be used to breach the security of a semiconductor device?

There are two basic classes of tampering techniques, non-invasive attacks and invasive attacks. In invasive attacks, a device is first decapped and then microprobed. Focused Ion Beams, or other techniques are used to try and determine the contents of the device. In non-invasive attacks, the security of the device is probed by external means such as brute force key generation or changing voltages to discover hidden test modes, etc.

How secure are Actel's Antifuse FPGAs from non-invasive attacks? Actel's antifuse devices are nonvolatile, which means the devices can be configured before they are shipped to the end-user. Unlike SRAM technology, there is no bitstream that can be intercepted, or external configuration device that can be compromised.

How secure is ProASIC$^{\underline{PLUS}}$ against non-invasive attacks?

The key on ProASIC$^{\underline{PLUS}}$ ranges from 79 to 263 bits in length. A brute force attack to randomly discover the key would require each key is tried in succession. It is not possible to load the key via the JTAG port at greater than 20MHz. This method would therefore require billions of years to discover the key. It is not possible to trick a ProASIC$^{\underline{PLUS}}$ part into a test mode by varying programming voltages or sending a specific code sequence.

How secure are Actel's Antifuse FPGAs from invasive type attacks?

Determining the state of an antifuse is exceedingly difficult. Antifuse-based FPGAs use a small piece of dielectric, usually smaller than 1µ square, as an open switch between two metal lines. Where a connection between two metal lines is desired, a programming pulse is used to short out the dielectric. This short is less than 100 nano-meters in diameter. These shorts are not visible when viewed from the top. Therefore, in order to physically identify them, it is necessary to de-process or cross-section the devices. Rather than being a precise method, this involves trial and error and typically requires that several cross sections be completed just to find a single link shorting out the dielectric.

How secure is ProASIC$^{\text{PLUS}}$ From Actel against invasive type attacks?

Unlike an ASIC, decapping a ProASIC$^{\text{PLUS}}$ device reveals only the structure of the device not the actual contents. To reverse engineer a ProASIC$^{\text{PLUS}}$ FPGA a thief would have to determine whether a charge is present on each configuration transistor's floating gate. A thief would require access to a programmed ProASIC$^{\text{PLUS}}$ device, very sophisticated equipment, and considerable time. Once a thief succeeds in determining the overall transistor layout and the locations of the programmed transistors on the chip, there is still the need to translate that pattern back to a configuration bitstream that can be used to program another part to create a clone of the design. To reverse engineer the design is even more difficult; the thief must map the bit pattern into the physical structure of the device in order to generate a schematic of the part. In essence he must first reverse engineer the architecture of the ProASIC$^{\text{PLUS}}$ device and then map the various transistor states into it in order to fully understand the IP contained in the design. An invasive attack involves time-consuming, expensive and tedious work.

FlashLock

What is "FlashLock$^{\text{TM}}$?"

FlashLock is a security feature offered on all Actel ProASIC$^{\text{PLUS}}$ devices. It allows a user to read and write to a device but prevents all unauthorized users from doing so.

How is the FlashLock feature enabled?

The FlashLock feature is enabled using a private key set by the user. After the device has been programmed and verified the key is enabled. Once the user key in enabled, nothing can be done to the device without first unlocking the user key.

Can you alter or read back the contents of a ProASIC$^{\text{PLUS}}$ device once it is programmed?

Yes. Although nonvolatile, ProASIC$^{\text{PLUS}}$ is also reprogrammable, however, you must have the appropriate key or the FlashLock feature will prevent you from accessing the contents. You cannot read or alter the contents of a programmed ProASIC$^{\text{PLUS}}$ device without access to the user key.

Doesn’t the Flash-based ProASIC^{PLUS} family from Actel require a bitstream?

Because ProASIC^{PLUS} is based on nonvolatile, reprogrammable flash technology, the device is programmed once and stays programmed until the user wishes to change it. This contrasts with SRAM based FPGAs, which are programmed every time they are powered up. Once programmed, a ProASIC^{PLUS} device is live at power up. Since there is no external bit stream, there is nothing that can be copied.

What is the key length in ProASIC^{PLUS}?

The key size varies with the size of the device. Even the APA075 & APA150 have keys longer than 75 bits. This long key length makes it difficult to attack the key via external brute force techniques that try to guess the key.

Device	Key Size
APA075	79 Bits
APA150	79 Bits
APA300	79 Bits
APA450	119 Bits
APA600	167 Bits
APA750	191 Bits
APA1000	263 Bits

Are Flash-based CPLD designs secure?

CPLD-based designs tend to be very simple with only a few hundred gates, security is typically not much of a concern in “ glue logic” applications. With the smallest programmable logic devices it is often possible to reverse engineer a CPLD with simple functional analysis, making security concerns of little or no consequence. To improve overall system security, many designers integrate multiple low-density CPLD designs into a single high-security FPGA. This also allows designers to realize significant performance improvements and benefit from lower silicon and board costs.

How do SRAM FPGAs implement security?

In general, the SRAM FPGA companies have been silent on the subject of security. Faced with the ease with which a bitstream can be cloned, one vendor was prompted to publish a statement saying: “The best protection against a mindless copy is legal.” Though a dissenting opinion would affirm it is much cheaper to protect your design up front than trying to recover your losses through the courts.

Beyond last-ditch legal means, one SRAM supplier has proposed a security scheme to prevent configuration data from being cloned. Their scheme uses either a DES or a Triple DES algorithm to encode the bit stream. The appropriate keys must be loaded in a secure environment and require battery backup to preserve the contents. A key file is created and is used by the JTAG programmer to program the key in the FPGA. The bitstream is then downloaded to the FPGA but now it is an encrypted bitstream and cannot be read without the proper key. This technique effectively foils cloning, but the device is still susceptible to reverse engineering via invasive attacks.

The biggest disadvantages of this technique is that the on board memory in the FPGA must be continuously powered to maintain the key; loss of the battery will render the device unusable, because the FPGA will no longer recognize the encrypted bitstream. This is a cumbersome attempt to get a volatile technology to emulate a nonvolatile technology.

Although the supplier claims negligible battery drain, battery shelf lives are finite. From a system engineering perspective this solution requires additional components and board area. The design engineer has the choice of using battery clips, which are frequently failure prone, or soldering a battery in place, which improves reliability but makes replacement more difficult. In any case, the battery becomes a single point of failure for the system. If the battery power is lost even momentarily the device will no longer function and must be returned to the OEM for reprogramming. This creates significant support issues for the end user.

Another more subtle issue involves the loss of control of the key. If an OEM chooses to use a contract assembly house he must also provide the assembler the key. Since the SRAM FPGA requires a constant battery supply to keep the key alive it cannot be pre-configured and must be initialized at board assembly. This places the key in a potentially non-secure environment. This makes it more likely that unauthorized users can obtain the key and the design integrity will be compromised.

Why is ProASIC$^{\underline{\text{PLUS}}}$ an effective solution for protecting sensitive IP?

ProASIC$^{\underline{\text{PLUS}}}$ offers a unique set of attributes unmatched by any other solution:

- Flash-based process technology and the unique ProASIC$^{\underline{\text{PLUS}}}$ architecture results in the highest security of any FPGA available at no incremental cost
- The only FPGA that is both nonvolatile and reprogrammable, the first viable ASIC alternative
- Because there is no external bit stream, no boot prom is required, saving valuable board space, lowering system cost, and eliminating a security risk
- Live at power up with no latency, just like an ASIC
- Built-in, not added-on, security features require no battery and eliminates reliability and serviceability issues

Why is Actel's antifuse technology more secure than comparable FPGA solutions available in the market today?

Industry experts consistently regard Antifuse as the most secure technology for PLD designers concerned with protecting their IP. The reality is there are several factors that serve as an impediment to those attempting to gain access to the contents of an Actel device. Fundamentally, the sheer number of antifuses in a typical Actel FPGA compared to the low percentage that are actually programmed make it virtually impossible to access the design using any type of invasive method. The SX72A device from Actel has over 6 million antifuses, and the state of each individual fuse has to be known to effectively reverse engineer the part. Furthermore, Actel's Silicon Explorer, an internal-node inspection tool, does not offer any risk, as some SRAM development tools, that might tip off the savvy criminal as to the location of key design elements and their associated bitstream.

What is FuseLock™ from Actel?

FuseLock is a security feature offered on all of Actel’s secure metal-to-metal antifuse FPGAs. FuseLock gives designers the capability to prevent unauthorized users from reading back the contents of an Actel FPGA.

Does Actel implement additional security measures in its antifuse architecture to thwart would-be attackers?

Yes. In addition to the inherent security characteristics of Actel’s proprietary antifuse architecture, additional steps have been taken to ensure even those with an understanding of the programming process can not compromise a design or gain access to the contents of the FPGA. The programming process is broken down into several steps that limit the ability for anyone with even partial knowledge of the device architecture and programming procedures to gain access to any portion of usable code**.**

5172063-0/8.02