Electronic Design

HIREN PARMAR, FPGA Security Vulnerabilities and Countermeasures,
eInfochips Inc. – An Arrow Company, https//www.einfochips.com/

FPGA Security Vulnerabilities and Countermeasures

This article explores the numerous risks associated with FPGA security and the recommended methods to keep the device secure.

A field-programmable gate array (FPGA), which consists of memory, programmable logic gates, and other components, is generally involved in digital-circuit design. FPGA settings are typically defined using hardware description languages (HDLs), such as VHDL and Verilog, much like that of application-specific integrated circuit (ASIC) configuration. You can modify the current configurations and add any new functionality or application requirements when needed.

FPGAs can be used in many consumer applications like cameras, smartphones, autonomous vehicles, image and video processing, and security systems. In the corporate realm, FPGAs are widely used in various industries, including servers, medical electronics, and military equipment.

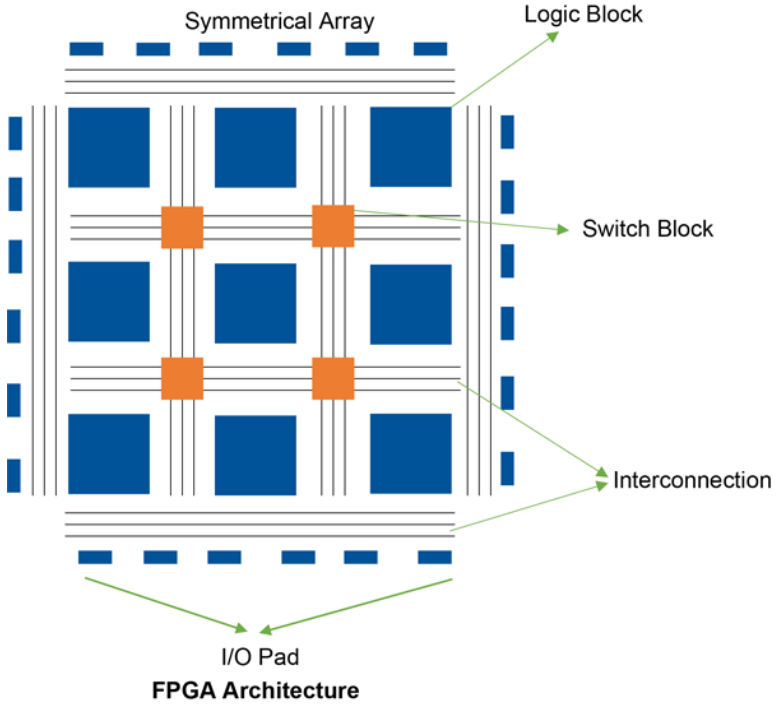
For instance, the aerospace industry implements FPGAs to manage anything from the Mars Rover to the Joint Strike Fighter. FPGAs also find homes in face-recognition systems, wireless network systems, intrusion detection systems, and supercomputers, which are all used in advanced security applications.

Approximately 80,000 separate commercial FPGA design projects started in 2005 alone. Moreover, the FPGA segment of global semiconductor market was valued at about $5.3 billion in 2021. Overall, the FPGA semiconductor market is set to reach a value of $9.3 billion by 2030, while growing at a compound annual growth rate (CAGR) of 6.5 percent during this forecast period *(Fig. 1)*.

Integrated Security Measures

Some built-in safety features are present in a well-designed FPGA. An FPGA is fundamentally less transparent than a conventional central processing unit (CPU). To build code and software that execute well, processors must have a well-documented instruction set, data pipeline, and memory architecture. With FPGAs, that's not the case.

The low-level functionality of FPGAs is formulated by the



1. Shown is a basic FPGA architecture with the symmetrical arrays, interconnections, logic blocks, and switch blocks.

developer, keeping them undocumented and thus creating a murky environment, making it harder to identify flaws. The mountain of paperwork involved makes it considerably more difficult to breach and infiltrate FPGAs, though it's still doable.

Risks to FPGA Security

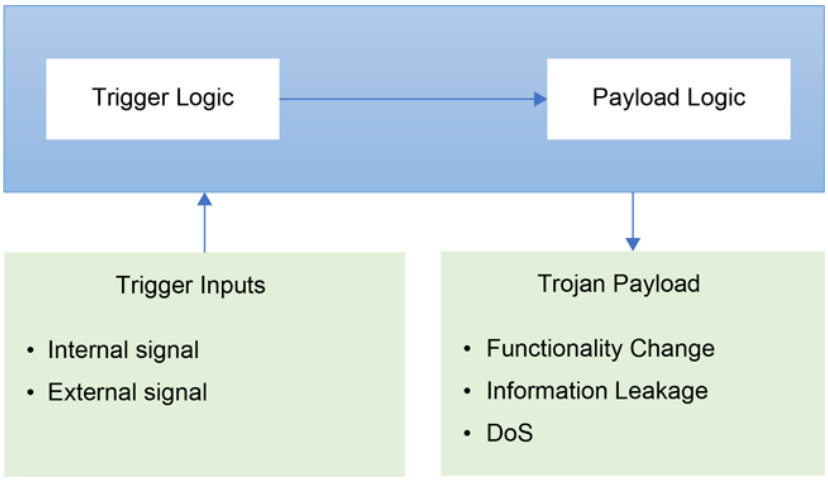
Intellectual property (IP) theft, harm to FPGA-based systems, and significant data loss are all associated with FPGA security threats. Security aspects are needed for each assault vary. The following categories can be used to divide the main FPGA attacks.

Cloning Attacks

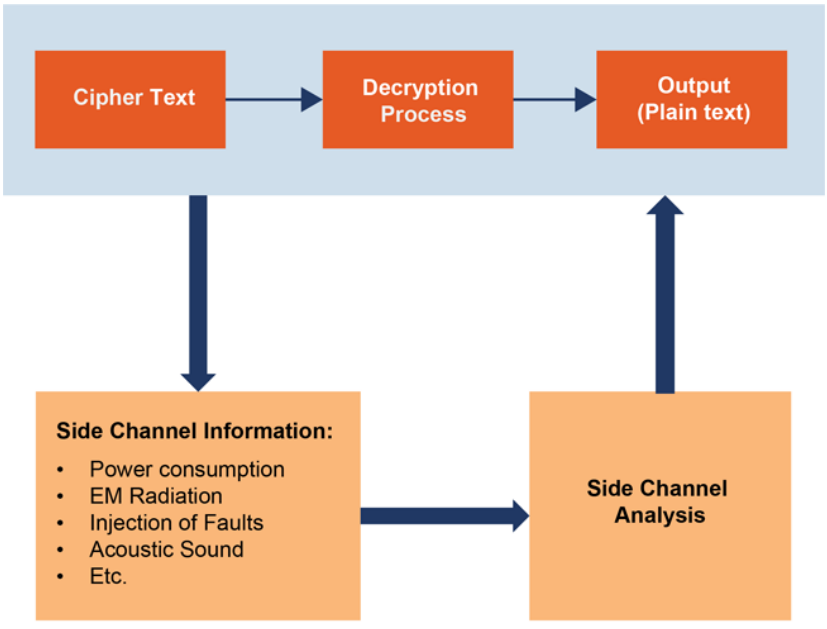
Attackers replicate FPGA development programming during cloning. They then utilize the bitstream in a similar gadget and market it as their own. Cloning may involve the complete design or just a portion of it. For instance, the seller may have restrictions on the purchased cores. It's a volatile FPGA's most typical security flaw.

Overbuilding

Overbuilding could result in an unreliable foundry manufacturing more FPGA chips than necessary and selling them to system developers for less money.



2. A hardware trojan affects physical circuits and can change a system's behavior.



3. This is how side-channel attacks turn the informational patterns of the system against it.

Hardware Trojans

Trojans are made to maliciously modify physical circuits and change a system's behavior. They harm hardware dependability, cause system failures, provide remote access to hardware, and pose a risk to sensitive data *(Fig. 2)*.

Side-Channel Attacks

Cybercriminals don't use conventional methods to break into the FPGA using side-channel attacks. Instead, they turn the informational patterns of the system against it. Side-channel attacks utilize the physical data that's exposed when a system is using an encryption technique. For instance, when a bitstream file is encrypted, which is supported by most FPGA vendors, side-channel attacks can leak the keys kept in the FPGA chips and render the bitstream unprotected *(Fig. 3)*.

Fault injection is the most common side-channel attack. Hackers introduce errors to test the system's response and then may create controllable flaws to modify the FPGA from that point. These assaults use voltage, timing, and laser faults. To find these patterns of information, the hacker typically must be nearby or in physical possession of the device.

Replay Attacks

A major security and privacy risk for FPGA design is the FPGA replay attack, which involves an attacker downgrading an FPGA-based system to an earlier version with known flaws *(Fig. 4)*.

Reverse Engineering

Gate-level netlist reverse engineering

and reverse engineering using image processing are the two basic types of IC reverse engineering. Adversaries can extract higher-level functionality from the gate-level netlist using gate-level netlist reverse engineering, such as register-transfer-level (RTL) or structure-level description.

Once they intercept the bitstream, hackers can employ reverse-engineering strategies to explore the FPGA further. There are tools specifically designed for mapping bitstream bits, recovering circuit designs, and other tasks. Although it isn't technically hacking, reverse engineering all or part of a bitstream is stealing IP from the creators *(Fig. 5)*.

Spoofing

The attacker's bitstream is substituted for the original FPGA bitstream during spoofing *(Fig. 6)*. That bitstream may contain elements obtained by reverse engineering or cloning. As a result, the system may become vulnerable, giving hackers effective control of the machine or system.

Such conduct could result in injuries or deaths brought on directly or indirectly by the hacker's actions in some safety-critical applications. A major security risk exists if the bitstream could be viewed remotely.

Tampering

Tampering involves attackers altering the application's design. The attacker can disable some features of the application or introduce logic that leaks data from it through tampering. Since tampering necessitates setting of values in the bitstream, it also can be considered reverse engineering.

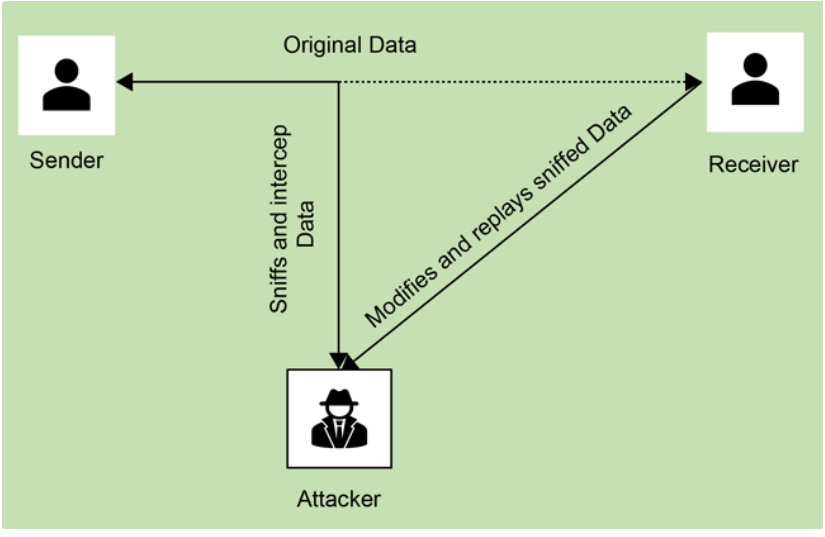
Bitstream Interception

Bitstream interception is one of the most frequent methods used by attackers to impact FPGAs. This security hole has lots of documentation. As far as vulnerabilities go, gaining access to those crucial configuration files unleashes a can of worms. Hackers can use files to seize control, steal bitstream data, and other methods.

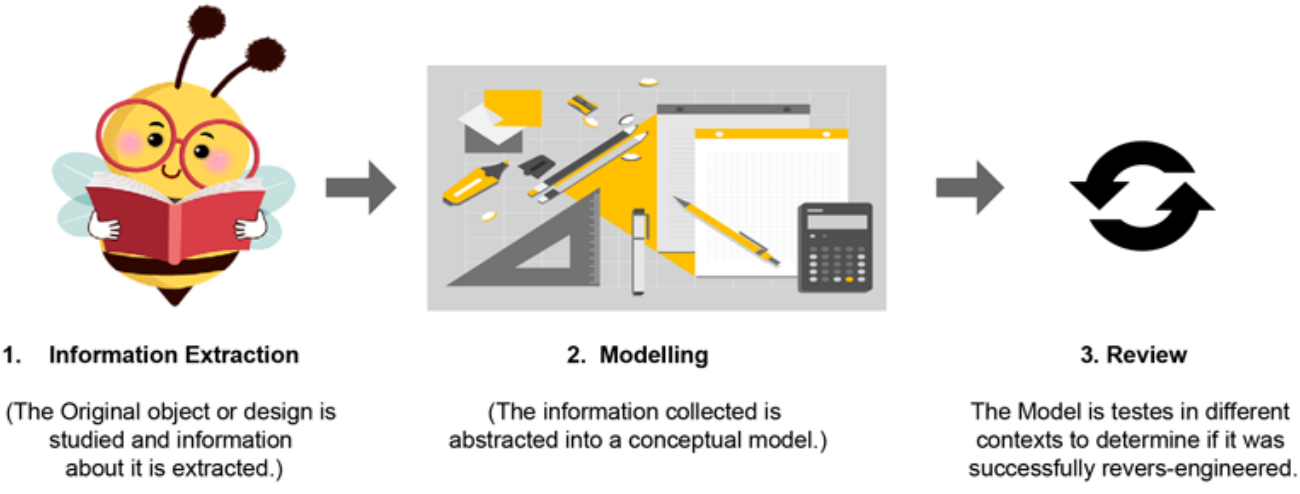
One of the most important jigsaw pieces to this puzzle is the bitstream. Once they have it, criminals are free to cause mayhem. To obtain the bitstream, hackers typically need physical access to the device.

Thermal Laser Stimulation

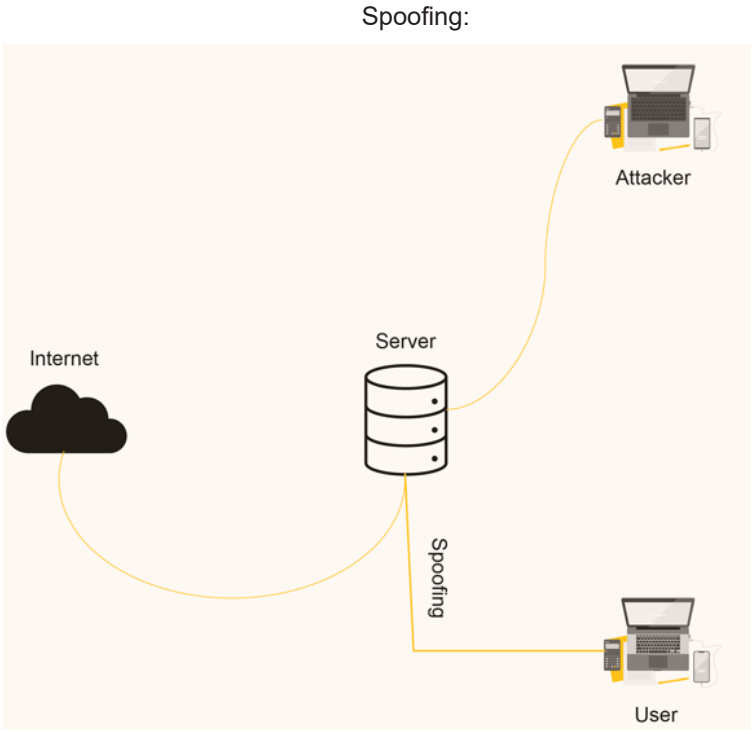
Thermal laser stimulation (TLS) is frequently used for fault analysis. This method also can be applied to identify and read a chip's memory contents with the goal of obtaining private information, like the key for bitstream decoding. It hasn't yet been established whether this attack method works against contemporary ICs that have



4. Replay attacks downgrade an FPGA-based system to an earlier version with bugs.



5. This is how reverse engineering steals IP from FPGA creators.



6. Attackers replace their own bitstream with the original FPGA bitstream during spoofing.

software composition analysis (SCA) defenses.

TLS attacks demand pricey hardware and extremely lengthy execution times (a professional microscope for failure analysis is necessary for this type of attack). Manufacturers of programmable logic devices, however, can't afford to overlook this category of attacks because the attack can be carried out even while the component isn't powered.

Security Solutions for FPGAs

On-chip security is made possible by an FPGA's' programmability, but this malleability also creates certain vulnerabilities.

Bitstream Encryption

The FPGA bitstreams must be encrypted and authenticated correctly. Effective encryption methods can stop side-channel attacks, data interception, and more. A volatile key is used in the finest kind of encryption for FPGAs. Like bitstream data, these keys remain in battery-baked RAM (Random Access Memory).

Keep your data encrypted, as decryption doesn't take place until after it's used and removed from SRAM. Data is heavily protected at every stage of the procedure. Cryptographic data with the volatile key is lost during a system power cycle.

The session keys used in this method of encryption are different each time. Hackers can't enter the system using side-channel attacks or other types of interception methods.

Isolation of the Configuration Process

FPGAs use on-chip isolation techniques to protect the system from microprocessor attacks. A comparable strain is placed on the FPGA since connected microprocessors are particularly susceptible to security problems.

The usual data routes are kept separate from the configuration procedure to prevent interference. It functions as a firewall and alters the attack surface. Added security is provided by the isolation, which also makes sure that the circuit can't change while being used.

Cyclic Redundancy Checks and Monitoring

Cyclic redundancy checks (CRCs) are able to find mistakes, unintentional corruption, and other unexpected issues. During transmissions, you can verify the bitstream using the CRCs to look for mistakes or deliberate alterations. During boot-up, logic analyzers can check the communication between the flash memory and the FPGA. They also will spot unusual Joint Test Access Group (JTAG) data and issues with other debug ports, which is helpful.

External Safety Devices

FPGAs can use external security systems to store encryption keys. For verification, the FPGA employs a challenge-response mechanism. The FPGA is given access once the external device knows the proper response.

Watermarks and Identifiers

Cloning and overbuilding can be stopped via watermarking and distinct digital identities. Developers can embed these independent distributors (IDs) in many ways. They may happen at the level of behavior, a netlist, a physical object, or even a bitstream.

Obfuscation

Obfuscation decreases reverse-engineering risk. It's intended to conceal a design's functionality by introducing seemingly random combinational logic gates. The technique secures the FPGA structure to make it even more challenging for hackers to decode by making it complex.

Hiren Parmar is an Engineer at eInfochips in the Cybersecurity domain. He specializes in IoT/Cybersecurity. He's an enthusiastic cybersecurity researcher who works to produce detailed and informative content on the subject. He has expertise in Hardware Security, Embedded Security, Side Channel Analysis, web application Vulnerability Assessment & Penetration Testing (VAPT), System Hacking, IOT/ Automotive security, Vulnerability Management, Threat Modelling, Cryptography, and Secure Software Development Life Cycle (secure SDLC).