National Security Agency
Cybersecurity Technical Report

DoD Microelectronics:

Field Programmable Gate Array

Overall Assurance Process

December 2022

U/OO/230112-22
PP-22-1374
Version 1.0

For additional information, guidance or assistance with this document, please contact the Joint Federated Assurance Center (JFAC) at https://jfac.navy.mil.

Notices and history

Document change history

Date	Version	Description
December 2022	1.0	Initial Publication

Disclaimer of warranties and endorsement

The information and opinions contained in this document are provided "as is" and without any warranties or guarantees. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government, and this guidance shall not be used for advertising or product endorsement purposes.

Publication information

Author(s)

National Security Agency
Cybersecurity Directorate
Joint Federated Assurance Center

Contact information

Joint Federated Assurance Center: https://jfac.navy.mil
General Cybersecurity Report Inquiries: CybersecurityReports@nsa.gov
Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov
Media inquiries / Press Desk: Media Relations, 443-634-0721, MediaRelations@nsa.gov

Purpose

This document was developed in furtherance of NSA's cybersecurity missions. This includes its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense information systems, and the Defense Industrial Base, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Executive summary

This document describes the overall process for implementing the Field Programmable Gate Array (FPGA) Assurance Policy for the DoD.

This document also serves as a guide for implementing an assurance platform for hardware assurance.

The overarching assurance process is summarized as follows:

- Standardize practices
- Detect and prevent threats
- Understand and respond to attacks

This document identifies standard practices, as well as the methods to use to evaluate any particular mitigation designed to detect and prevent threats.

Contents

Table

1 Standardize practices

The goal of the Levels of Assurance process is to develop a robust set of standards and guidance that captures how programs select and achieve the desired level of assurance (LoA). For the field programmable gate array (FPGA) effort, this guidance was captured in the following six documents, available to the DoD at https://jfac.navy.mil:

- *FPGA Overall Assurance Process*
- *Levels of Assurance Definitions and Applications*
- *FPGA Threat Catalog*
- *FPGA Level of Assurance 1 Best Practices*
- *FPGA Level of Assurance 2 Best Practices*
- *FPGA Level of Assurance 3 Best Practices*

In any assurance process, it is essential that the methodology is consistent and repeatable. The three levels of assurance (LoA), their mitigations, and the metrics to measure their effectiveness are designed to be vendor and product independent. They are designed to provide mitigations at the highest level of granularity where they are effective and target the threats that originate from a malicious actor. This actor has malicious intent and could have capabilities ranging from the level of a commercial enterprise up to that of a well-resourced nation-state effort.

In developing standards and best practices, the assurance team performing this work should be able to:

- Identify and define specific LoAs. Using the LoAs established in *FPGA Overall Assurance Process* is strongly recommend for all hardware problems. Different processes may be appropriate in the software arena where inexpesive or easier to implement threats are more practical.
- Establish concrete criteria to determine what attacks must be mitigated to achieve each LoA.
- Identify the threats of interest and bundle them in groups with common mitigations.
- Determine which threat category is relevant to each LoA.
- Use the Attack-Countermeasures Analysis (ACMA) tree approach to validate the assumptions made and ensure the thoroughness of the mitigation approach.

These areas of effort are further defined in the following sections with a summary of how each was addressed in the FPGA arena.

1.1 Identify and define specific levels of assurance

The LoAs are designed to protect hardware systems in programs that have varying assurance requirements. Standards and guidelines need to be designed to support these multiple assurance levels. In most cases, the standards and guidelines will need to accommodate both high and low LoAs.

The LoAs defined by the FPGA team were intended to apply across the hardware space. The number of levels corresponding to the extremes must be decided. It is essential that the number of levels remain small. A guiding assumption of this process is that it is effectively impossible to measure the difference in importance between two threats with extreme accuracy. While a broad categorization is possible, attempts to provide too much granularity will make it extremely difficult to agree on threats of concern or mitigation effectiveness.

In the end, the levels should provide the appropriate protections without imposing unnecessary costs on programs. In the case of FPGA Assurance, three levels accounted for the needs of projects utilizing FPGAs.

- LoA1 - Threats that are inexpensive to implement but with high consequences.
- LoA2 - Threats that have greater expense and investment but are less targetable or most likely conducted for pre-positioning.
- LoA3 - Threats that are very expensive to implement or have unpredictable outcomes.
- These levels are further detailed in the *Levels of Assurance Definitions and Applications* document.

1.2 Establish concrete criteria

Once the assurance team defines the specific LoAs, they must outline a set of concrete criteria by which a threat may be evaluated to determine which LoAs must mitigate it. This criteria must be as objective as possible and enable the team to assess each threat through a simple set of questions. Additionally, the criteria does not necessarily need to limit a threat to a single assurance level for mitigation as there may be multiple

ways to carry it out. However, the circumstances that place the threat in its respective LoA should be clear.

In the case of FPGA LoAs, the criteria was based on the cost to implement a threat and the usefulness of the attack. More details regarding these criteria may be found in the *Levels of Assurance Definitions and Applications* document. The intent here was to avoid threat rankings based on small variations in an attack, but rather to use simple questions about how each threat can be carried out and the effects they would have.

The required access, technology, and investment needed to carry out an attack would stand as criteria used to measure its cost. The value of effect criteria measures the positive outcome of the attack for an adversary, given success. The targetability criteria measures the utility of an attack. Each criteria has been designed, as much as possible, to be a straightforward question with limited ambiguity. These criteria serve not only as a guide for measuring attacks, but also mitigations. To be of value, a mitigation does not need to be perfect, but it must have a measurable effect on one of the criteria significant enough that an attacker must take additional actions covered by a higher LoA. Table I provides a description of the threat cost and impact at each LoA level.

Table I: Example FPGA LoA Table

	LoA1 **Inexpensive, significant consequences**	LoA2 **More expensive, moderate effects**	LoA3 **Expensive, unpredictable outcome**
Threat Effort or Cost			
Access	A single available point of access	A difficult point of access *or* multiple available points of access	Multiple points of difficult access
Technology	Existing public technology	Low implementation risk technology	Technologically feasible
Investment	Minimal investment of resources	A large multidisciplinary team	A nation scale and directed priority
Threat Consequences or Impact			
Value of Effect	Disable or subvert a system	Establish vulnerabilities (for future exploitation)	Degrade system performance
Targetability	Inherently targetable and useful	Affect only a subset of systems	Blind attacks (difficult to precisely target or control the outcome)[1]

1.3 Identify the set of known threats of interest

The next step in the assurance process is to develop the set of threats of interest and bundle them in groups with common mitigations expected to address them. Those threats should be explored ensuring the entire attack space is identified. Each of the

[1] LoA3 systems are best approached with a full risk analysis to identify which blind attacks are of concern to a given system. Realistic risks in this space are often idiosyncratic, and the most concerning blind attacks are typically not the most expensive. LoA3 is the least "one size fits all" of all the categories specifically because many such systems are judged to need to concern themselves with such unpredictable effects.

attacks should be identified and documented into a threat catalog. This process is expert driven and it is strongly recommended that, if possible, the assurance team has expertise in the technology, as well as operations in the human, cyber, and supply chain domains.

To explore the attack space, the team should begin with a brainstorm identifying all possibilities. This should consider the complete supply chain. The team should think through all the possible attacks that could happen, including those that are already mitigated.

In the FPGA Assurance development process, the FPGA Trust Study developed by Sandia National Labs was used as a major component of this. A study of that level of detail may not be necessary in all cases, but the complete supply chain for the technology of interest must be considered. In many ways, this is an exercise in expert development.

Next, these attacks should be condensed into a smaller set. Attacks should not be removed, but similar attacks at similar places in the supply chain should be grouped. That is, attacks which the experts believe are likely to be mitigated by the same approach should be under the same category.

For instance, in the FPGA Trust Study there is a threat "Adversary modifies GOTS system at design." This attack includes hacking into a contractor network, but it also includes paying off a contractor on the design team to do something. It even includes breaking and entering into the contract design facility. The commonality is that all of it is happening during the design process at a U.S. government contractor site and in a place where there is access to design code. As a result, these threats can be captured under a single threat.

1.4 Determine to which level of assurance each threat is relevant

Topic experts should measure each threat by the established assurance criteria and map them to their appropriate level of assurance for mitigation.

For the FPGA Assurance Trust Study, each threat was assigned to two experts and measured against the cost and utility criteria. Any points of disagreement were discussed until resolved or raised up to the greater group. The goal is not to make

comparisons between threats or decide which is more important, but to decide which threats meet the criteria at a given level of assurance and categorize them at that level.

The goal of the entire assurance process is to provide honest feedback about the vulnerability of a supply chain, regardless of the cost of implementation or ability to mitigate any particular threat. The goal is **not** to provide a list of the “top five” or “best bang for your buck” mitigations. If cost/benefit decisions need to be made, they should be made separately from the assurance process. Even threats that are known from the outset to be too costly or technically infeasible to address must still be included in the assessment in order to present a holistic view of the threats and enable decision makers to make fully informed decisions.

1.5 Identify the common mitigations against the threats

The next step is to evaluate known mitigations as follows:

1. **Mitigation proposal** - In mitigation proposal, a mitigation is identified and defined.

   In the FPGA Assurance Trust Study, this was a brainstorming session in which experts identified what mitigations exist. Additional inputs to this are research or development efforts that are preparing for tech transition. An identified mitigation should be associated to the threats which it may mitigate.

2. **Mitigation vetting** - In mitigation vetting, an expert performs a "sanity check" to determine plausibility. The purpose of this check is to determine if a mitigation would make a difference assuming it is as effective as it could be.

   To be valid a solution, a mitigation needs to increase one of the five criteria for evaluating an attack by at least an LoA step as represented in Table I. **A valid mitigation will make one of the criteria so much more difficult to carry out that the threat category will become at least one LoA higher.** For instance, a mitigation may make it impossible for a single compromised insider to carry out the threat, moving the “access” criteria from LoA1 to LoA2. In other words, a mitigation is of interest if it makes an attack within the threat harder in a way that is measurable by the criteria, and that attack was not already measurably harder than other attacks in the threat.

3. **Mitigation evaluation** – In mitigation evaluation, a mitigation that has survived vetting must be evaluated. While vetting assumes that the mitigation is as

effective as it claims or could be, the evaluation determines how effective the mitigation actually is.

As mitigations receive positive evaluations, they are grouped into mitigation packages. Mitigation packages come in two forms: standard packages and custom packages. The intent is that most programs seeking a low level of assurance will be able to use a standard package. Higher levels of assurance may require a custom package.

A standard package is a pre-developed set of mitigations that achieve a specific level of assurance when implemented. They must be developed by a team of experts that is independent of implementation and is qualified to evaluate threats. The package identifies a set of mitigations that are achievable and cost efficient for the given level of assurance, and includes them in a single document. Hardware vendors may perform some of these mitigations. For instance, in FPGA Assurance, some key elements of assurance happen in the design and manufacture of the FPGA itself. Other mitigations must be implemented by the program at design, manufacture, or through a device's lifecycle.

A program that has issues with the standard packages may create a custom package. A custom package mitigates the same threats, and must use mitigations approved through the same process. But it may choose to implement different mitigations that are more cost efficient or operationally convenient for that program. Over time, recurring custom packages will lead to the development of new standard assurance packages.

1.6 Work with vendors and stakeholders

Of vital importance to the development of assurance practices is the involvement and partnership with the various stakeholders. These stakeholder groups should include the DoD JFAC labs and DoD program elements. Working together, these groups will better define what levels of assurance are needed, what the threats are, what mitigations exist, and which mitigations need to be developed. In addition, input from relevant hardware vendors may help to inform the development in terms of existing threats and mitigations.

Additionally, collaboration among all the stakeholders allows the early adoption of assurance practices since the concerns of all parties are known from the beginning. In the case of FPGA Assurance, this collaboration was conducted through the FPGA Working Group (FPGAWG) of the National Defense and Industry Association (NDIA).

This group established a set of guidelines using existing mitigations, first while working on a more comprehensive package, and then in parallel with its release. The involvement of all the stakeholders allowed for this. This collaborative team also reviewed all products coming from the effort for completeness and applicability.

Finally, specific areas were also identified where the hardware vendors could provide material assistance in improving the current FPGA assurance environment. Together, the collaboration resulted in documented assurance practices and policies that had the support of all stakeholders.

2 Prevent and detect threats

Following the development of policy and best practices, two things must follow:

- An evaluation of the effectiveness of the recommended mitigations must be conducted.
- Research and development of new mitigations to close existing threat gaps and to maximize the efficiency of those already in place. This effort should focus on two areas:
    - Threats that do not have current or cost-effective mitigations, and
    - Threats that do not have mitigations at higher levels of assurance.

The key to evaluating the mitigations is identifying an appropriate organization to evaluate the mitigation that is sufficiently independent from its implementation. Not all mitigations are technical. To be successful, an assurance program needs multiple organizations available to evaluate multiple kinds of mitigations. These include technical mitigations and more traditional counter-intelligence mitigations.

Whatever organization is doing the evaluation should be independent of the organization that developed the mitigation. Evaluations can be compromised by conflicts of interest when the evaluators' management or the evaluators themselves are invested in the success of the mitigation. As a rule, the results of the evaluation should not be subject to review by anyone who is in the chain of command of the developer. In addition, the evaluators should not consult on the development of the mitigation.

The evaluation team will determine if the mitigation:

- Fully mitigates a threat,

- Could mitigate a threat in combination with other mitigations, or
- Fails to mitigate the threat in a substantial way.

The evaluation team should create a set of criteria that defines the limits and conditions in which the mitigation is effective. This should include the level of monitoring or checking appropriate to validate that the mitigation is employed correctly.

In the development of assurance policies, the work done to collect threats and mitigations will identify areas in which either no mitigation exists, no comprehensive mitigation existed, or only expensive complex ones are available. Research should be focused on closing these gaps.

In the case of FPGA Assurance, the Broad Agency Announcement (BAA) process was used heavily to collect proposals for solutions to areas where there were no effective mitigations. The idea was to complete the research and solution development in a year and then release it for use throughout the DoD. This process illustrates that the assurance process will evolve and require updates with the changing threats and mitigations.

3 Respond to detections

An assurance process needs to be responsive to changes in adversary attacks. As such, it needs capability and infrastructure to detect new threats and then to evaluate them. Finally, this evaluation should result in actionable information that can be used to create new mitigations.

In the hardware realm, these tasks require laboratory capabilities. In the cases where gaps in the capability exist, research should be targeted in developing detection and analysis techniques. Additionally, it will be necessary to fund any measures to bring the current lab capabilities in line with the evaluation needs.

4 Use the assurance process

A DoD program with hardware assurance concerns can rely on their assurance process as a guide to mitigate attacks at the appropriate level. When applying these practices, each program should keep the following in mind:

- The program office must determine the appropriate assurance level that applies to their project. *DoD Microelectronics Levels of Assurance Definitions and*

Applications contains guidance for determining the appropriate level of assurance for a system and its components. The concepts presented there can be directly applied to any area of hardware assurance.

- Once the level of assurance is chosen for a project, that program should merge the associated mitigation plans into its own Program Protection Plan as soon as possible.
- The hardware in question will remain assured at the LoA chosen as long as the recommended mitigations are in place.
- In cases where a program can demonstrate that compromised hardware does not result in a project vulnerability, the program would not have to qualify for an LoA.
- In cases where the program has developed custom mitigations, they should have a defined path for approval of their approach.

Hardware assurance is not something that is added onto a project at the end. It needs to be planned for from the very beginning as it will have impact in the areas of:

- Hardware acquisition strategy,
- Design development environment and personnel vetting,
- Network security,
- Software validation and use,
- Design testing,
- Reliability and product testing, and
- Product assembly.

In the effort to stand up the FPGA Assurance Process, all of these factors were taken into account and supported.

5 Conclusion

This document captures the process used by FPGA Assurance in implementing a hardware assurance process. This resulting guidance can be used across DoD, or this process can be used allowing for program customization. For questions or inquiries contact JFAC at https://jfac.navy.mil/.

Appendix A: Standardized Terminology

The following terms are used in the Joint Federated Assurance Center Field Programmable Gate Array Best Practices documents. These terms are modified from Defense Acquisition University definitions to support common understanding.

Application design – The collection of schematics, constraints, hardware description language (HDL), and other implementation files developed to generate an FPGA configuration file for use on one or many FPGA platforms.

Application domain – This is the area of technology of the system itself, or a directly associated area of technology. For instance, the system technology domain of a radar system implemented using FPGAs would be "radar" or "electronic warfare."

Configuration file – The set of all data produced by the application design team and loaded into an FPGA to personalize it. Referred to by some designers as a “bitstream”, the configuration file includes that information, as well as additional configuration settings and firmware, which some designers may not consider part of their “bitstream.”

Controllable effect – Program-specific, triggerable function allowing the adversary to attack a specific target.

Device/FPGA device – A specific physical instantiation of an FPGA.

External facility – An unclassified facility that is out of the control of the program or contractor.

Field programmable gate array (FPGA) – In this context FPGA includes the full range of devices containing substantial reprogrammable digital logic. This includes devices marketed as FPGAs, complex programmable logic devices (CPLD), system-on-a-chip (SoC) FPGAs, as well as devices marketed as SoCs and containing reprogrammable digital logic capable of representing arbitrary functions. In addition, some FPGAs incorporate analog/mixed signal elements alongside substantial amounts of reprogrammable logic.

FPGA platform – An FPGA platform refers to a specific device type or family of devices from a vendor.

Hard IP – Hard IP is a hardware design captured as a physical layout, intended to be integrated into a hardware design in the layout process. Hard IP is most typically distributed as Graphic Design System II (GDSII). In some cases, Hard IP is provided by a fabrication company and the user of the IP does not have access to the full layout, but simply a size and the information needed to connect to it. Hard IP may be distributed with simulation hardware description language (HDL) and other soft components, but is defined by the fact that the portion that ends up in the final hardware was defined by a physical layout by the IP vendor.

Level of assurance (LoA) – A Level of Assurance is an established guideline that details the appropriate mitigations necessary for the implementation given the impact to national security associated with subversion of a specific system, without the need for system-by-system custom evaluation.

Physical unclonable function (PUF) – This function provides a random string of bits of a predetermined length. In the context of FPGAs, the randomness of the bitstring is based upon variations in the silicon of the device due to manufacturing. These bitstrings can be used for device IDs or keys.

Platform design – The platform design is the set of design information that specifies the FPGA platform, including physical layouts, code, etc.

Soft IP – Soft IP is a hardware design captured in hardware description language (HDL), intended to be integrated into a complete hardware design through a synthesis process. Soft IP can be distributed in a number of ways, as functional HDL or a netlist specified in HDL, encrypted or unencrypted.

System – An aggregation of system elements and enabling system elements to achieve a given purpose or provide a needed capability.

System design – System design is the set of information that defines the manufacturing, behavior, and programming of a system. It may include board designs, firmware, software, FPGA configuration files, etc.

Target – A target refers to a specific deployed instance of a given system, or a specific set of systems with a common design and function.

Targetability – The degree to which an attack may have an effect that only shows up in circumstances the adversary chooses. An attack that is poorly targetable would be more likely to be discovered accidentally, have unintended consequences, or be found in standard testing.

Third-party intellectual property (3PIP) – Functions whose development are not under the control of the designer. Use of the phrase “intellectual property”, IP, or 3PIP in outlining this methodology of design review does not refer to property rights, such as, for example, copyrights, patents, or trade secrets. It is the responsibility of the party seeking review and/or the reviewer to ensure that any rights needed to perform the review in accordance with the methodology outlined are obtained.

Threat category – A threat category refers to a part of the supply chain with a specific attack surface and set of common vulnerabilities against which many specific attacks may be possible.

Utility – The utility of an attack is the degree to which an effect has value to an adversarial operation. Higher utility effects may subvert a system or provide major denial of service effects. Lower utility attacks might degrade a capability to a limited extent.

Vulnerability – A flaw in a software, firmware, hardware, or service component resulting from a weakness that can be exploited, causing a negative impact to the confidentiality, integrity, or availability of an impacted component or components.

Appendix B: LoA1 Mitigation Overview

The following are the “top takeaways” from provided recommendations. Full details on implementation can be found in the JFAC FPGA best practices documentation.

- Design Process
  - Apply robust NIST approved cybersecurity processes to development environments.
  - Perform regular design and code reviews with multiple people involved at each step.
  - Perform robust testing with complete requirements coverage and high code coverage.
  - Use a reproducible build process to generate FPGA bitstreams/configurations.
  - Apply asymmetric authentication algorithms to all configuration files before system start. Manage the private keys using an HSM.
- Design IP and Tools
  - Obtain design software from reputable sources. Validate before installation that its hash is as expected.
  - Select third-party IP carefully, and do not accept encrypted or obfuscated IP blocks. Either notify JFAC about its use or evaluate it for suitability through a security audit.
- Parts Acquisition and Assembly
  - Acquire parts through reputable channels. Validate parts’ authenticity either cryptographically or physically.
  - Assemble the system in a controlled way, or validate afterwards that it was assembled correctly. Ensure that the correct configuration and keys are installed.
- Reporting
  - Assist JFAC by providing high-level information about the program’s FPGA usage.
  - Contact JFAC if there is suspicion of technical interference by an adversary.