Analysis of FPGA Development Processes and Associated Security Threats

Minseok Hur[1*[0009-0008-8889-7516]], Jiho Shin[2*[0009-0000-1855-4747]], Moohong Min[3**[0000-0001-8979-1344]], and Aram Kim[4**[0000-0003-1030-3534]]

[1] Department of Immersive Media Engineering, Sungkyunkwan University, Humanities and Social Sciences Campus, 25-2, Sungkyunkwan-ro, Jongno-gu, 03063 Seoul, South Korea
alexhur3535@skku.edu

[2] Department of Forensics, Sungkyunkwan University, Humanities and Social Sciences Campus, 25-2, Sungkyunkwan-ro, Jongno-gu, 03063 Seoul, South Korea
sing0021@g.skku.edu

[3] Department of Computer Education/Social Innovation Convergence Program, Sungkyunkwan University, Humanities and Social Sciences Campus, 25-2, Sungkyunkwan-ro, Jongno-gu, 03063 Seoul, South Korea iceo@skku.edu

[4] Department of Information Security, University of Suwon, 17, Wauan-gil, Bongdam-eup, Hwaseong-si, Gyeonggi-do, South Korea aramkim@suwon.ac.kr

[*] These authors equally contributed to this paper

[**] Corresponding Author: Moohong Min, Aram Kim

Abstract. Field-programmable gate arrays (FPGAs) are versatile integrated circuit that can be programmed and reconfigured to perform a wide range of tasks. Their flexibility and simplicity, in comparison to traditional application specific integrated circuits (ASICs), have led to widespread adoption in various domains requiring high performance and real-time processing such as embedded systems, robotics, and artificial intelligence. However, numerous known vulnerabilities and security threats associated with these devices have been identified. This study investigates the safety development and verification and validation (V&V) methodologies for FPGAs, with a focus on safety requirements applicable to nuclear domain. Furthermore, it provides a comprehensive analysis of security threats and vulnerabilities across the lifecycle of a FPGA. The study emphasizes that traditional development models are insufficient for addressing known and potential risks, highlighting the necessity for an integrated approach that combines safety and security considerations in FPGA development.

Keywords: FPGA, Vulnerability, Security, Safety, Lifecycle

1 FPGA: Overview and Security Threats

1.1 Overview of FPGA

Field-programmable gate arrays (FPGAs) are utilized in a multitude of fields, including embedded systems, robotics, power electronics, drive applications and artificial intelligence-based industrial control systems [1]. FPGAs are versatile, user-programmable

devices comprising an array of configurable logic cells embedded in a connection matrix [2]. The cells can be programmed to perform a variety of logic functions. The design process comprises a series of steps, including mapping, placement and routing (PAR), with the objective of optimizing the configuration. FPGAs permit the concurrent execution of multiple operations and are frequently employed in conjunction with computer-aided design (CAD) tools to facilitate the implementation of complex designs. These characteristics render FPGAs suitable for a plethora of applications, from the initial prototyping phase to the deployment of final products. In contrast to systems that necessitate an operating system, FPGAs implement solely the requisite functionality, thereby optimizing system performance and reducing complexity. Furthermore, FPGAs exhibit characteristics that are analogous to those of software during the development phase but are analogous to those of hardware during the execution phase.

The fundamental components of an FPGA are programmable logic blocks, programmable routing that connects these logic functions, and input/output (I/O) blocks that interface with external connections [3]. The logic block is comprised of various elements, including look-up tables (LUTs), flip-flops (F/Fs), and multiplexers (MUXs). The interconnect network, which connects these logic functions, facilitates the seamless transmission of data and signals through routing channels, switch boxes, and connection blocks. I/O blocks facilitate the interfacing between the FPGA's internal logic and external circuits.

In order to implement a design on an FPGA, it is necessary for the user's hardware description language (HDL) source code to undergo a series of transformations that produce a configuration executable by the FPGA [4]. The final stage of the design implementation process is the generation of a bitstream, which is a binary file containing the specific configuration data required to program the FPGA. The bitstream is essentially a sequence of bits that directly configures the FPGA's logic blocks, interconnects, and I/O cells, thereby enabling the device to perform the desired functionality as described in the HDL. As illustrated in Figure 1, the generation of a bitstream involves several key steps, including synthesis, mapping, placement, routing, and finally bitstream generation. Each of these steps is crucial in ensuring the FPGA operates as intended in the final application, delivering the desired performance and functionality.

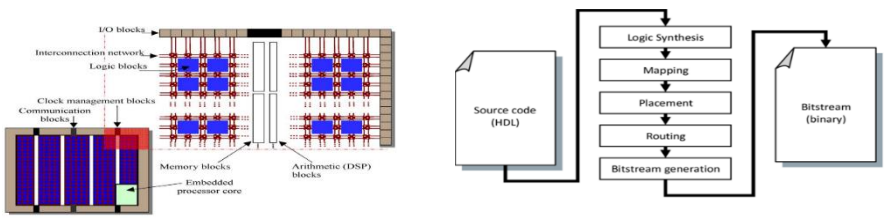

Fig. 1. Generic FPGA Architecture [1] and Bitstream Generation Process [4]: The left image shows the layout of logic blocks, interconnection resources, and I/O cells in an FPGA, while the right image illustrates the steps to generate a bitstream from HDL source code.

2 Safety Development Process for FPGA

2.1 FPGA Development and V&V Process

FPGAs are being used in critical environments including nuclear power plants where the highest levels of safety and security are imperative. NuScale Power LLC, a company that specializes in the development of small modular reactors (SMRs) has developed a highly integrated protection system platform (HIPS) utilizing FPGAs [5, 6]. Similarly, in South Korea, Korea Hydro & Nuclear Power Company (KHNP) is planning to obtain approval for standard design for i-SMR, which incorporates FPGAs in its safety systems, by 2030 [7].

However, FPGAs may be vulnerable to cyberattacks, including those involving physical access, due to existing known vulnerabilities. Furthermore, the effective integration of safety and security is essential. Therefore, it is crucial to develop and analyze a comprehensive development process that ensure safety and security of FPGAs.

The IEC 62566 standard outlines a verification and validation (V&V) process to be followed throughout the development life cycle of HDL-programmed devices (HPDs) such as FPGAs. This standard specifies the necessary requirements to be met for HPDs [8]. The process considers both the hardware features of HPDs, including synthesis and place-and-route (PAR), as well as HDLs such as VHDL [9] or Verilog [10]. The V&V process is crucial in ensuring that the system is reliable and secure [11-13]. An overview of the V&V process throughout the lifecycle is shown in Fig. 2. Analysis of the activities and the requirements for verification for each development phase is presented in the following sections.

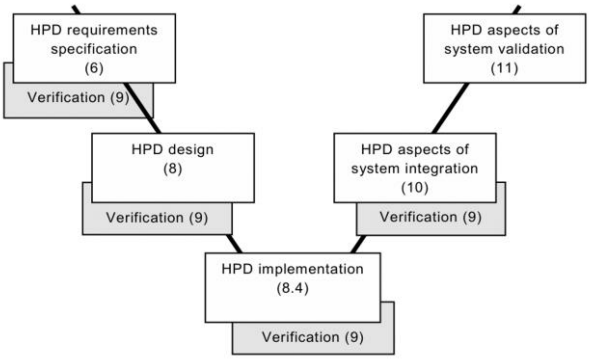

Fig. 2. Development Life Cycle of HPD. (The numbers refer to the section of IEC 62566 [8]).

2.2 Requirements Specification Phase

HPD requirements specification. This phase specifies the requirements of the functions, modes, environments including timings and electrical characteristics of the input

and output sequences of the HPD. Fault detection and fault tolerance is also addressed for a defensive design.

Verification. A critical analysis of the requirements must be performed and documented to avoid inconsistency, omissions and ambiguities. Abnormal behaviors such as unexpected input values or sequences needs to be checked.

2.3 Design and Implementation Phase

HPD design. The design of the HPD must produce a formalized description of HDL such as register transfer level (RTL) and its associated documents. Fault detection and behaviors corresponding to the specified requirements are needed. It should follow a set of strict design and coding rules, a top-down approach, and a synchronous architecture for stable and robust design.

HPD implementation. In this phase, a gate-level description (netlist) of the HPD, which is the result of synthesizing RTL, is used to perform PAR, which results in the physical description, such as the programming file or bitstream, needed to produce the HPD. Timing information (back-annotation) and files of parameters and constraints needed for synthesis and PAR must be produced. Results of the post-route analysis, finite state machine analysis, and static timing analysis (STA), including timings and electrical characteristics, must be documented.

Verification. Both phases must be reviewed to examine the completeness, correctness, and robustness. The review team must include hardware experts and engineers from teams responsible for the system or components of the HPD.

2.4 Testing Phase

HPD aspects of system integration. Assembling and interconnection of verified hardware and software components are done to build intermediate and final targets. The plan for this phase must be prepared in the design and implementation phases and should include the sequence and timings for both input and output signals and the acceptance criteria. All requirements of the HPD must be verified and tested. Fault resolution procedures must also be applied.

Verification. Verification of components and subsystems is done to determine whether they are properly integrated into the system and performs as specified.

HPD aspects of system validation. System validation involves simulating every function in normal operations, anticipated occurrences, and accident conditions using static

and dynamic input signals. It also includes documenting HPD results, configurations, and tests, allowing external reviews.

3 Security Threats to FPGA

3.1 Security Vulnerabilities Across FPGA Lifecycle

Despite the implementation of rigorous safety protocols, there is a possibility that potential security vulnerabilities may still emerge in FPGAs. While numerous studies have explored these vulnerabilities, the principal security threats to FPGA systems can be classified as follows:

Table 1. Security Vulnerabilities in the FPGA Development Process.

Stage	Security Vulnerability	References
Design	Inadequate FPGA design	[14][15]
Implementation	Theft and cloning of bitstreams	[16][17]
Deployment	Malicious bitstreams	[18][19][20]
Operation	Side-channel attacks	[21][22][23]
Design & Implementation	The presence of backdoors and Trojans in the design	[24][25]
Deployment & Operation	Supply chain attacks	[26][27]

These threats manifest throughout the design, implementation, deployment, and operation phases of FPGA systems.

During the design phase, vulnerabilities such as insufficient encryption, lack of verification, and the insertion of backdoors and Trojans present a significant threat to the security of the system. These issues have the potential to significantly impair the overall reliability of the system, as they may allow the covert insertion of malicious functions or channels into the hardware components during this phase.

In the implementation phase, the primary security concerns are insufficient encryption, lack of verification, and the insertion of backdoors and Trojans. It is possible for an attacker to illegally copy or steal FPGA configuration files, thereby compromising the intellectual property and system security of the relevant entity. Furthermore, the introduction of malicious bitstreams into an FPGA can result in the disruption of normal functionality or even the malfunction of the device.

In the deployment and operation phase, the primary security concerns are side-channel attacks and supply chain attacks. Side-channel attacks exploit the physical operations of the FPGA, including power consumption, electromagnetic emissions, and timing variations, to extract sensitive information such as encryption keys. Supply chain attacks entail the introduction of malware by malevolent insiders or external entities during the manufacturing process or along the supply chain, thereby compromising the FPGA's integrity and reliability.

4 Conclusion

This study highlights the critical importance of integrating safety and security considerations into the FPGA development process, especially in safety-critical environments such as nuclear power plants. While FPGAs offer several advantages, they are vulnerable to various security threats, including bitstream theft, side-channel attacks and the insertion of malicious backdoors. Our analysis reveals that traditional safety regulations are insufficient for addressing these risks, necessitating the need for a development model that integrates safety and security from the requirements specification phase to the system validation phase. The identified vulnerabilities illustrate the necessity for a comprehensive approach that systematically addresses potential threats at every stage of the FPGA development lifecycle. Future research needs to focus on refining this approach to ensure secure deployment of FPGAs in safety-critical environments.

Acknowledgements

This work was supported by the Nuclear Safety Research Program through the Korea Foundation Of Nuclear Safety(KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission(NSSC) of the Republic of Korea. (No. RS-2024-00403596)

References

1. Monmasson, E., Idkhajine, L., Cirstea, M. N., Bahri, I., Tisan, A., Naouar, M. W.: FPGAs in industrial control applications. IEEE Transactions on Industrial Informatics 7(2), 224–243 (2011).
2. Brown, S. D., Francis, R. J., Rose, J., Vranesic, Z. G.: Field-programmable gate arrays, vol. 180. Springer Science & Business Media, Heidelberg (2012).
3. Farooq, U., Marrakchi, Z., & Mehrez, H.: Tree-based heterogeneous FPGA architectures: application specific exploration and optimization. Springer Science & Business Media. (2012).
4. Arcas Abella, O.: Beehive: An FPGA-based multiprocessor architecture. Master's thesis, Universitat Politècnica de Catalunya (2009).
5. NuScale Power: Submittal of the approved version of NuScale Topical Report TR-1015-18653, "Design of the Highly Integrated Protection System Platform," Revision 2 (CAC No. RQ6005) (2017).
6. US NRC: Safety evaluation by the office of new reactors licensing topical report (TR)-1015-18653-P (Revision 1) "Design of Highly Integrated Protection System Platform," NuScale Power, LLC, Docket: PROJ0769 (2016).
7. Choi, S.: Current Situation of Innovative SMR Instrumentation and Control Design. In: Nuclear Safety & Security Information Conference 2023, pp. 1–6. Nuclear Safety & Security Information Conference (2023).
8. IEC 62566-2012: Nuclear power plants – Instrumentation and control important to safety – Development of HDL-programmed integrated circuits for systems performing category A functions (2012).

9. IEEE 1076-2019: IEEE Standard Criteria for VHDL Language Reference Manual (2019).
10. IEEE 1364-2005: IEEE Standard for Verilog Hardware Description Language (2005).
11. IEEE 1012-2004: Software Verification and Validation (2004).
12. Wu, Y., et al.: Development, verification and validation of an FPGA-based core heat removal protection system for a PWR. Nuclear Engineering and Design 301, 311–319 (2016).
13. Maerani, R., Mayaka, J. K., Jung, J. C.: Software verification process and methodology for the development of FPGA-based engineered safety features system. Nuclear Engineering and Design 330, 325–331 (2018).
14. Chaudhuri, S.: A security vulnerability analysis of SoCFPGA architectures. In: Proceedings of the 55th Annual Design Automation Conference, pp. 1–6. ACM (2018).
15. Wanderley, E., Vaslin, R., Crenne, J., Cotret, P., Gogniat, G., Diguet, J. P., ... & Torres, L: Security fpga analysis. Security Trends for FPGAS: From Secured to Secure Reconfigurable Systems, 7-46. (2011).
16. Hutchings, D., Taylor, A., Goeders, J.: Toward FPGA Intellectual Property (IP) Encryption from Netlist to Bitstream. ACM Transactions on Reconfigurable Technology and Systems (2020).
17. Zhang, J., Qu, G.: Recent attacks and defenses on FPGA-based systems. ACM Transactions on Reconfigurable Technology and Systems (TRETS) 12(3), 1–24 (2019).
18. Ender, M., Swierczynski, P., Wallat, S., Wilhelm, M., Knopp, P. M., Paar, C.: Insights into the mind of a trojan designer: The challenge to integrate a trojan into the bitstream. In: Proceedings of the 24th Asia and South Pacific Design Automation Conference, pp. 12–119. ACM (2019).
19. Chakraborty, R. S., Saha, I., Palchaudhuri, A., Naik, G. K.: Hardware Trojan insertion by direct modification of FPGA configuration bitstream. IEEE Design & Test 30(2), 45–54 (2013).
20. Ender, M., Moradi, A., Paar, C.: The unpatchable silicon: A full break of the bitstream encryption of Xilinx 7-series FPGAs. In: 29th USENIX Security Symposium (USENIX Security 20), pp. 803–1819. USENIX (2020).
21. Moradi, A., Barenghi, A., Kasper, T., Paar, C.: On the vulnerability of FPGA bitstream encryption against power analysis attacks: Extracting keys from Xilinx Virtex-II FPGAs. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 111–124. ACM (2011).
22. Glamočanin, O., Coulon, L., Regazzoni, F., Stojilović, M.: Are cloud FPGAs really vulnerable to power analysis attacks?. In: 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1007–1010. IEEE (2020).
23. Giechaskiel, I., Eguro, K., Rasmussen, K. B.: Leakier wires: Exploiting FPGA long wires for covert-and side-channel attacks. ACM Transactions on Reconfigurable Technology and Systems (TRETS) 12(3), 1–29 (2019).
24. Krieg, C., Wolf, C., Jantsch, A.: Malicious LUT: A stealthy FPGA Trojan injected and triggered by the design flow. In: 2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–8. IEEE (2016).
25. Skorobogatov, S., Woods, C.: Breakthrough silicon scanning discovers backdoor in military chip. In: Cryptographic Hardware and Embedded Systems–CHES 2012, pp. 23-40, Springer (2012).
26. Miller, J. F.: Supply chain attack framework and attack patterns. The MITRE Corporation, MacLean, VA (2013).
27. Thrangrycat, https://thrangrycat.com/, last accessed 2024/08/13.