National Security Agency
Cybersecurity Technical Report

DoD Microelectronics: NSA Evaluation of Commercially Available Embedded FPGAs

NOVEMBER 2023

U/OO/232094-23
PP-23-3645
Version 1.1

For additional information, guidance, or assistance with this cybersecurity technical report (CTR), please contact the Joint Federated Assurance Center (JFAC) at JFAC_HWA@radium.ncsc.mil.

Notices and history

Document change history

Date	Version	Description
October 2023	1.0	Initial publication
November 2023	1.1	Minor revisions

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

NSA Joint Federated Assurance Center: JFAC_HWA@radium.ncsc.mil

Cybersecurity Report Feedback / General Cybersecurity 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

An embedded Field Programmable Gate Array (eFPGA) is a custom-sized piece of configurable FPGA fabric that has been designed for use in an application specific integrated circuit (ASIC). Integrating this intellectual property (IP) into an ASIC enables a program to optimize their system to meet product requirements while providing the ability to reconfigure portions of the design after manufacture in the same way that an FPGA can be reprogrammed. At the time of the evaluation effort, four different commercial vendors were examined: Achronix, Flex Logix, Menta, and QuickLogic.

The NSA JFAC Hardware Assurance (HwA) evaluation team conducted a five-phase evaluation of the eFPGA commercial offerings. These five phases and their reports are named Rocky I-V and include:

1. **Rocky I** – report on eFPGA vendors, their products, their business models, design flows, the advantages of the using this IP type, and the application spaces where this product can have the most impact.
2. **Rocky II** – report on designing custom-built eFPGA blocks and evaluating their deliverables, documents, and support.
3. **Rocky III** – report on the back-end software support for developing and loading applications into the eFPGA in the ASIC hardware.
4. **Rocky IV** - report summarizing the security-related benefits and vulnerabilities of using an eFPGA.
5. **Rocky V** – report on actual ASIC development effort using eFPGA.

Following the completion of these research phases, the NSA HwA evaluation team made many observations regarding the eFPGA vendors, the eFPGA products, the vendor support software, deliverables, integration, programming, and security/assurance issues. These observations are as follows:

- The four vendors evaluated have mature eFPGA products from commonly used foundries/processes. These products have been used in many DoD systems.
- The eFPGA products are well supported by the necessary software and electronic design automation (EDA) models required for use in an ASIC development flow.

- An eFPGA provides a good assurance solution for protecting sensitive ASIC design information from disclosure during the manufacturing of ASICs.
- The eFPGA products are complicated and require some additional work to integrate into an ASIC compared to other commonly used hard IP.
- The NSA JFAC HwA evaluation team believes that this product set is ready to be used by DoD.

This summary report ends with a list of recommendations to the U.S. Government (USG) and to potential users of eFPGAs. They are summarized as follows:

- The eFPGA should be viewed primarily as an ASIC IP block and not as an FPGA substitute.
- The USG should engage with eFPGA vendors to communicate USG assurance and security needs and to influence product design and development in a manner that would benefit DoD programs.
- The USG should recognize eFPGA as a means to maintain confidentiality of sensitive portions of DoD-specific ASIC designs during manufacture. This fact makes the continuing maturity of this product a DoD ASIC assurance priority.
- Since eFPGA macros are custom IP blocks, they represent a target of opportunity for an adversary to attack specific programs – their targetability is built in. Accordingly, DoD should engage JFAC to develop assurance best practice guidance to support this product.
- The USG should investigate the development of security functions and architectures to protect these devices and their bitstreams from malicious attacks. This could include government-off-the-shelf encryption and authentication engines to protect the configuration information containing sensitive design data.
- This report provides potential users with a list of detailed technical recommendations for selecting the correct eFPGA, developing a custom macro, performing acceptance testing on the deliverable, and conducting design integration.

This summary report reflects the state of eFPGA offerings available in the commercial market at the time of the evaluation effort, which began in 2020. These products may

have matured and been enhanced in the intervening time. The reader is encouraged to educate themselves on any relevant updates.

Providing many benefits to the ASIC designer, the current commercial offerings are widely available, well supported, reasonably mature, and powerful as an assurance tool. The NSA JFAC HwA evaluation team recommends the use of eFPGAs and believes that they can play an important role in protecting sensitive DoD algorithms.

Table of contents

List of figures

1. Introduction

For the last 20 years, the pace of technological advancement in microelectronics has gifted computer engineers a myriad of tools and intellectual property (IP) to spawn ever more advanced products. New fabrication processes, materials, memory devices, processors, and high-speed transceivers have all created new leaps in performance, miniaturization, and power reduction in custom DoD systems.

These advances include field programmable gate arrays (FPGAs), which have also grown in complexity. More than just the configurable microelectronic circuits of earlier generations, the current generation FPGAs now provide processor systems, a high level of security, high-speed input/output (I/O), and easy-to-use design environments. In many ways, FPGAs now provide much of what can be done on an application specific integrated circuit (ASIC) with the added ability to be reprogrammed.

In yet another advance, new ASIC IP has emerged on the commercial market known as the embedded FPGA (eFPGA). This new device enhances capabilities that can be obtained when designing a custom computer chip. An eFPGA is a custom-sized piece of configurable FPGA fabric that has been designed for use in an ASIC. Integrating this IP into an ASIC enables a program to optimize their system to meet product requirements while providing the ability to reconfigure portions of the design after manufacture in the same way that an FPGA can be reprogrammed. During the evaluation effort, the following four commercial vendors were examined:

Figure 1: Companies providing embedded FPGAs

2. Embedded field programmable gate array

An eFPGA is an IP block designed to enable changes to a portion of an ASIC's digital logic on demand. Normally, an ASIC is limited to performing only the functions for which it was designed. Its logic and functions are static and unchangeable. In this light, a computer chip that needs an updated function or a bug fix would require the design of an updated chip and a new manufacturing cycle to produce it.

Conversely, commercial FPGAs are microelectronic devices designed to have their functionality programmed into them after manufacturing, but at the cost of larger size, less performance, and greater power consumption. They are immune to the ASIC redesign problems because their logic is reprogrammable. As an example, if version A of a product needs a serial to parallel interface (SPI) and version B needs a universal asynchronous receiver/transmitter (UART) interface, the FPGA designer simply programs the same FPGA device to operate differently.

The eFPGA was developed to provide this same ability for an ASIC without sacrificing the optimized performance, area, and power that full custom chips provide. The eFPGA is a piece of FPGA programmable fabric that can be integrated into an ASIC layout and subsequently programmed in the same way that today's commercial FPGAs can. It is not a replacement for FPGAs in a DoD system, but a tool to increase the flexibility and usefulness of an ASIC design. This new IP is designed to work within industry standard design flows/tools, may be reprogrammed an unlimited number of times, and is configured with a bitstream in the same manner as commercial FPGAs.

Additionally, since the eFPGA is programmed after fabrication, it provides a mechanism for DoD programs to protect their sensitive algorithms and designs from disclosure during manufacturing, while still enjoying the benefit of optimization inherent in an ASIC. These can be programmed with sensitive code and functionality after the manufacturing process, thereby reducing the risk of compromise.

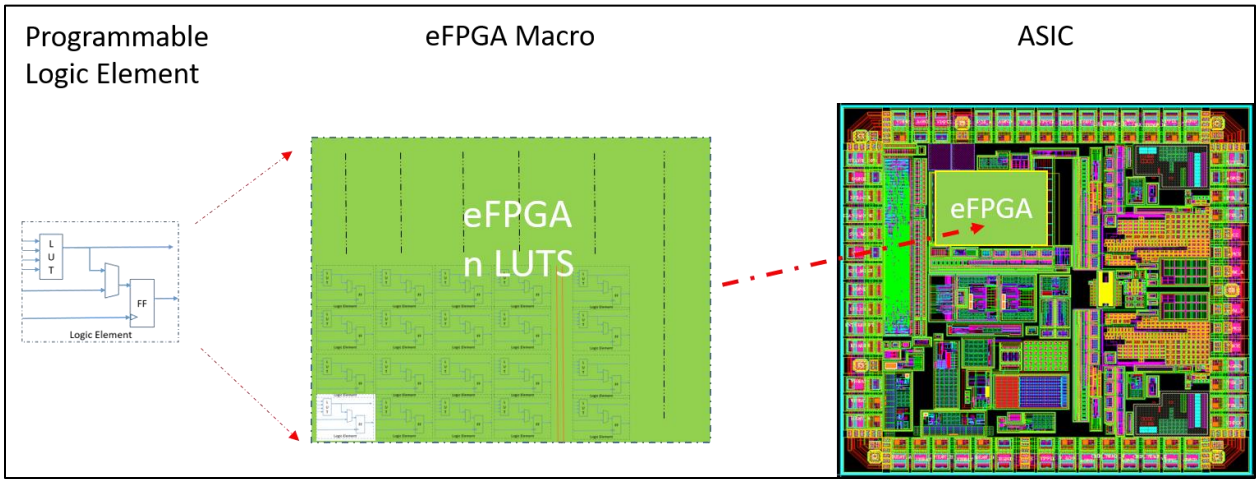

Figure 2: Logic Element (LE) to ASIC - the eFPGA is an array of programmable LEs forming an ASIC embeddable FPGA fabric

As depicted in Figure 2, currently all of the eFPGA manufacturers develop their IP from a basic building block called a programmable logic element (LE). This LE is tiled into a large field of programmable devices whose number is determined by the user. These LEs all include a proprietary look-up-table (LUT) function, registers, multiplexors, and programmable interconnection paths. Additionally, many of the eFPGA manufacturers offer options to integrate local memory, digital signal processors (DSP), and other functional features into their fabrics.

Each eFPGA block has hundreds to thousands of input and output pins allowing broad access to these capabilities. Configuration logic is provided with each eFPGA, either as an embedded function, or as a wrapper for the device. To support the designer, these blocks come with electronic design automation (EDA) views to assist with all of the common ASIC development phases, including: simulation, synthesis, timing analysis, test insertion, place and route, clock tree synthesis, post-route extraction, physical verification, and manufacturing tests. Furthering the support, eFPGA blocks are accompanied with advanced and easy-to-use software to assist the user with:

- Designing a custom eFPGA block
- Integrating it into an ASIC design
- Implementing a configurable digital function into the fabric
- Generating a bitstream for programming the device

These devices are referred to as “hard macros” in the ASIC world meaning they are delivered as completed computer-aided design (CAD) layouts as opposed to hardware description language (HDL) code that would require mapping to a physical library element. As such, they are foundry and process specific. This means an eFPGA block that was designed to work in a 32 nanometer (nm) process in foundry A, will require engineering work to make it work in a 14nm process in foundry B.

However, these devices are currently available in a wide range of technologies and each vendor is prepared to port their IP to new manufacturing processes where a viable business case exists.

3. Benefits and tradeoffs of using an eFPGA

eFPGAs provide many advantages to the ASIC designer, but with a few tradeoffs. The advantages are categorized into the following six primary areas:

- ASIC development risk mitigation
- ASIC lifecycle management
- Improvement in space, power, and performance (SWaP)
- System costs
- System security
- Securing sensitive algorithms from the ASIC manufacturer

Many of the advantages contribute in more than one of these areas and are referenced many times in this report.

3.1. ASIC development risk mitigation

An eFPGA is a powerful tool for mitigating risk in both design and fabrication. Today’s electronic designs demand greater performance at lower power, forcing programs to rely on advanced chip fabrication nodes. These cutting edge processes are complex and expensive, so correcting design issues can require second or third fabrication runs, which are both costly and time consuming. Therefore manufacturers are always seeking better ways to mitigate the risk of working with these complex technologies.

By using the new eFPGA IP, a designer can incorporate potentially risky or immature algorithms into the eFPGA fabric and evaluate them without first committing to an expensive and time-consuming manufacturing process, as shown in Figure 3. Using the

eFPGA allows the ASIC to be updated over the chip lifespan without the need for a new manufacturing cycle. Updates can include bug fixes or functional revisions. With costs of a new manufacturing cycle exceeding $10M for cutting edge process nodes, this ability represents significant cost savings and risk mitigation. While current mitigation strategies, such as multi-project wafers are effective, they can require more than nine months of production and manufacturing time per iteration.

Using the eFPGA reduces the need for additional fabrication runs, saving the time and costs of a new production run, thereby getting the product to market in less time for less cost. The eFPGA would also allow the program to support requirements for multiple functional variants of the ASIC. For instance, if the program has multiple customers with slightly different I/O configurations or different serial interface standards, the differences could be programmed into the eFPGA, requiring one manufacturing run and reducing program risk.

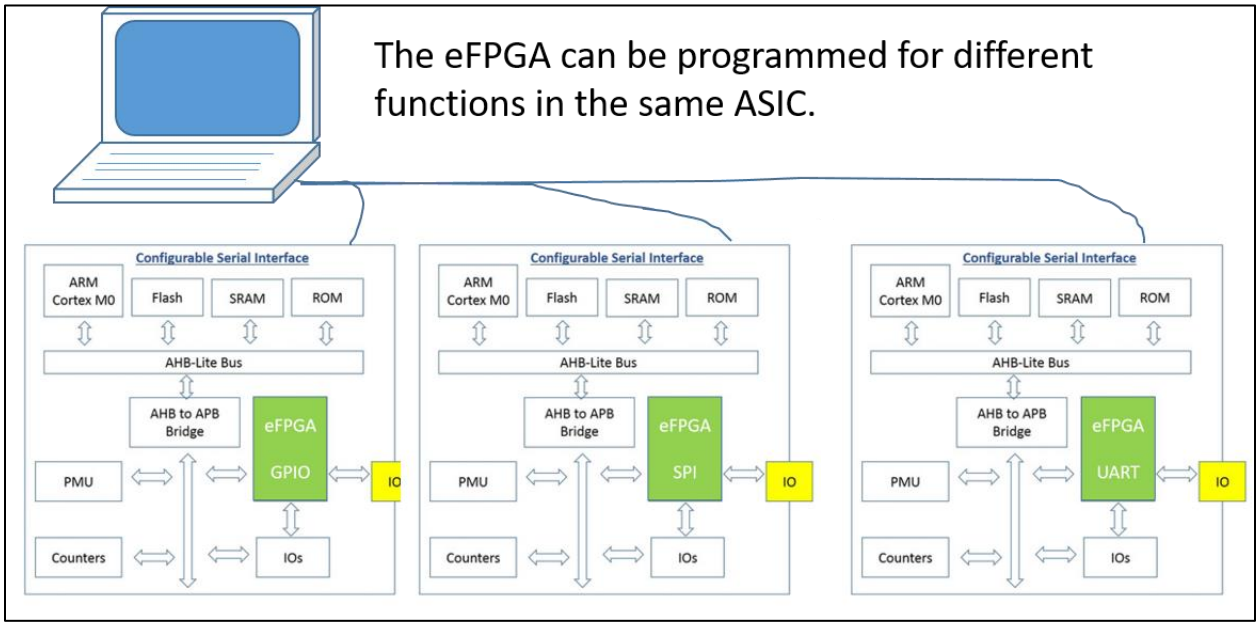

Figure 3: Same ASIC, different functions - the eFPGA provides an area on the chip that can be reprogrammed with different functions as needed

3.2. ASIC lifecycle management

Many ASIC products are planned as a series of releases, each with its own fabrication run to incorporate increasing functionality. In this scenario, an end user may have to update an IP function to a newer version or add new functionality in subsequent ASIC version releases. Historically, a version update used an existing FPGA or it needed an

ASIC redesign that would require a subsequent fabrication run, both of which would mean significant added costs and schedule delays.

However, incorporating an eFPGA into the ASIC design can help avoid this by allowing the designer to update existing hardware design or by adding new features via a new bitstream. This is illustrated in Figure 4. Additionally, this enables the designer to manage the new rollout schedule and avoid the cost of additional manufacturing runs/fabrication spins. In some cases, the designer may choose to include the eFPGA fabric in the ASIC production design to meet a future, currently unstated, need.

One of the vendors builds their eFPGA macro exclusively out of standard cell library elements, while the other three use a combination of standard cells and custom developed functions. Using standard cell libraries simplifies the porting effort of moving the product to a new manufacturing technology and speeds the process.

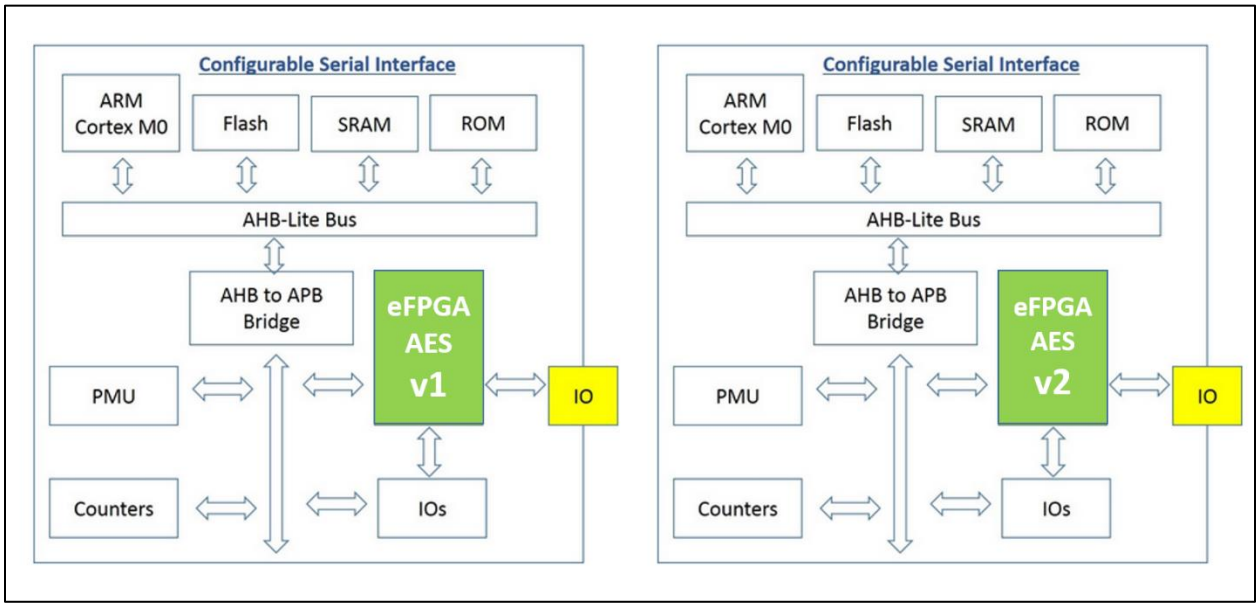

Figure 4: Lifecycle Management - the eFPGA provides the ability for a program to update versions of included functions in the future without refabricating the part

3.3. Improvement in space, power, and performance (SWaP)

An eFPGA can optimize an ASIC to enhance its performance, power, and space attributes. While a few of these optimizations are inherent to the eFPGA block itself, most of them exist because of the added configurability in the ASIC. The eFPGA can be generated using the standard cell library and threshold voltage (Vt) option that best fits the program needs. A programmable fabric could be custom made using a technology's library and optimizing its performance while still meeting design requirements, such as

lower power consumption. However, the eFPGA makes its greatest performance contribution when an on-board commercial FPGA can be replaced with an ASIC that incorporates an eFPGA. In this scenario, the ASIC can communicate directly with the fabric without going off-chip through an I/O bottleneck, see Figure 5. The communication between the ASIC and the eFPGA fabric can be pushed to the performance limits of the technology and across very wide parallel hardware buses. This also simplifies the clocking scheme, shortens the insertion delay on the clock tree, and eliminates the complexity of synchronizing off-chip devices with other logic on the ASIC. Finally, the system designer has the option of instantiating multiple fabrics on an ASIC, creating efficient architectures for specific applications.

The same manufacturing benefits can be used to create power and space efficiencies. The system will require less power because the commercial FPGA has been removed from the board. In addition, the high-speed transceivers providing communication between the ASIC and commercial FPGA may also be removed, yielding substantial power savings. The eFPGA blocks can all be generated in power-optimized libraries for lower leakage and most provide low-power states, such as sleep and power-down, further improving the power profile of the system. The eFPGA can also contribute to reducing the space needed for a system by removing the need to include a commercial FPGA on a board, freeing space for other beneficial applications.

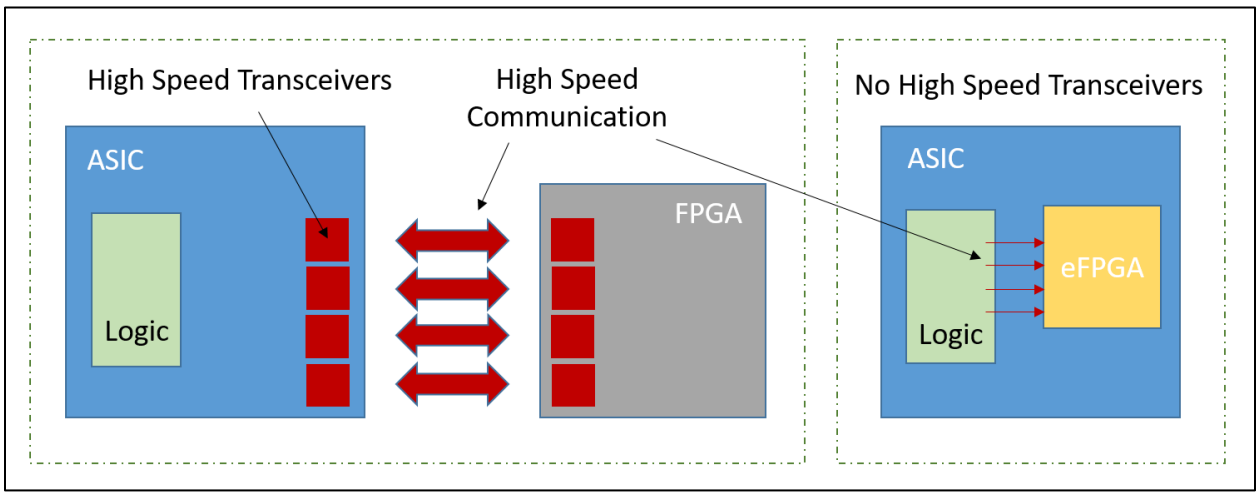

Figure 5: High Speed Transceivers - in many cases, moving the system FPGA fabric onto the ASIC can eliminate expensive, complex, and power-hungry high-speed transceivers

3.4. System costs

As stated earlier, incorporating an eFPGA into an ASIC can potentially result in fewer fabrication runs, saving costs associated with creating new mask sets to support design modifications or fixes. There are other potential mask related savings. If a program has different requirements from different customers, such as different package-to-pin signal assignments, functions, or I/O protocols, the only way to meet these requirements without incorporating an eFPGA into the ASIC design is by including all of the options on a single ASIC or by producing multiple ASIC versions. Both of these options are costly. By using an eFPGA, the designer can create multiple versions of an ASIC using a single mask set and tailoring the ASIC functionality to meet differing customer requirements.

In addition to the cost savings, this design approach can reduce design size since there is no longer the need to include every potential function or configuration option. This smaller design size lends itself to improved wafer yield at manufacturing – again, likely reducing cost.

When the eFPGA replaces a commercial FPGA, there are secondary cost savings. FPGAs require complex board designs and the requisite discrete support components, but an eFPGA reduces that complexity and those dependencies. The program can also save cost on IP license fees for using high-speed transceivers along with their associated integration and testing costs. Using a common eFPGA across multiple platforms means the license costs are amortized over a larger number of units, making each unit less expensive as more are built.

Finally, the eFPGA can deliver cost savings to a system that has large volume production. Commercial FPGAs do not have significant quantity discounts: the costs are similar regardless of purchasing volume.

For instance, purchasing 50,000 FPGA units at $1000/per unit would cost $50M. In contrast, 50,000 ASIC units with eFPGA in a cutting edge node would cost under $20M, and as illustrated in the following figure:

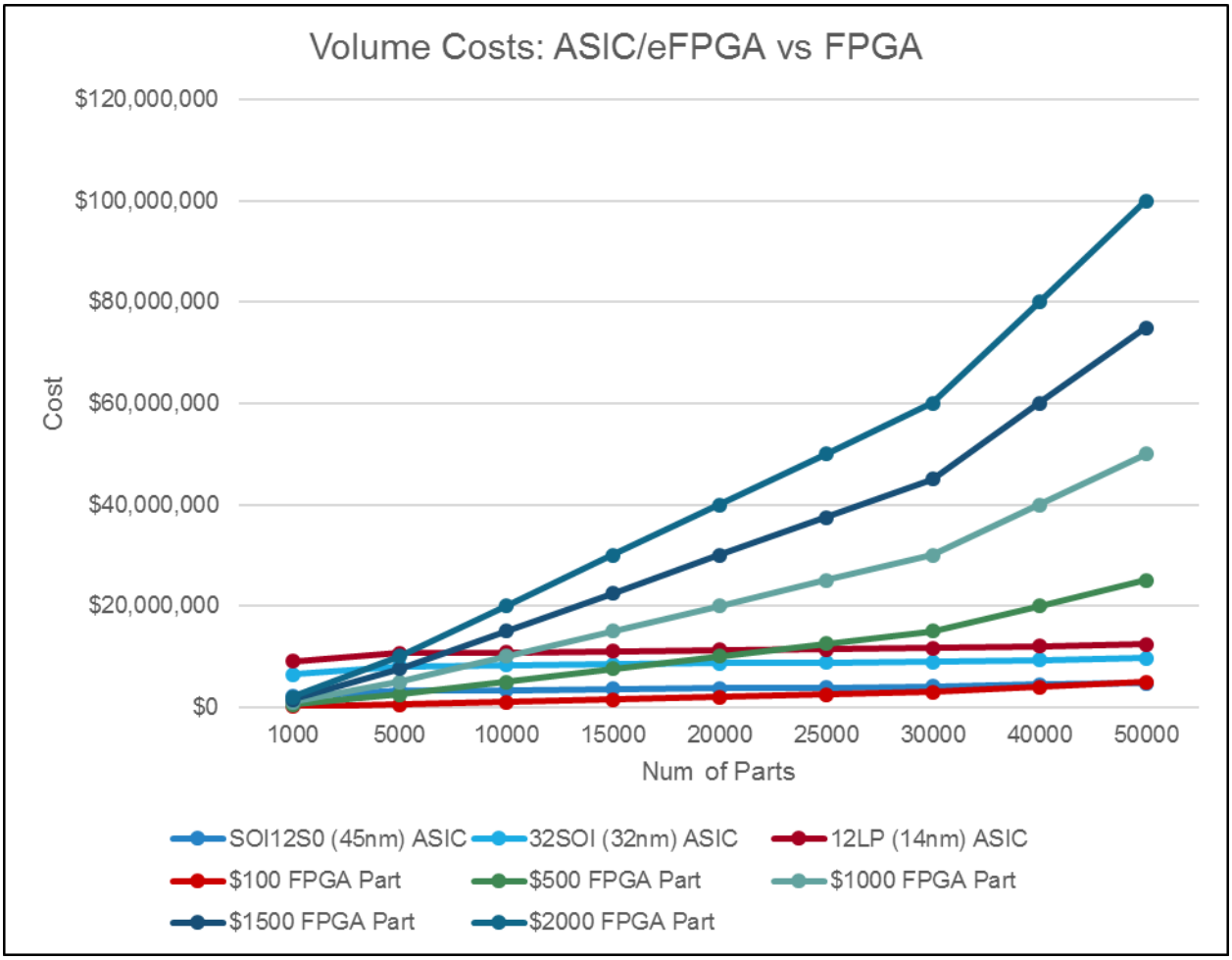

Figure 6: Volume Costs - Chart Assumptions – Approx. GlobalFoundries DoD Trusted Access Program Office wafer fabrication costs, 65% die yield on 12" wafers and $1.5M eFPGA license fee

3.5. System security

Incorporating an eFPGA into an ASIC design offers security benefits, but may also require a few tradeoffs. Unlike commercial FPGAs, the eFPGA is delivered without any security features since it is intended to be encapsulated within the anti-tamper boundary of the ASIC and reside under the system's overall security protections. This allows the designer to fully optimize the block and eliminate non-applicable anti-tamper features, saving surface area and power.

However, if the program requires dedicated or custom architected security, the burden of protecting the programming bitstream falls on the ASIC architecture. Under this setup, the designer only includes those protections that are needed. These protections could be at the module level, the board level, or at the chip level similar to a commercial

FPGA. While this model can provide additional engineering challenges in some architectures, it also provides a few unique opportunities. With the eFPGA, the designer is unencumbered by commercial-grade protection mechanisms and is free to implement any combination of bitstream obfuscation, encryption, and authentication to meet the level of security needed for each design as illustrated in the following figure.

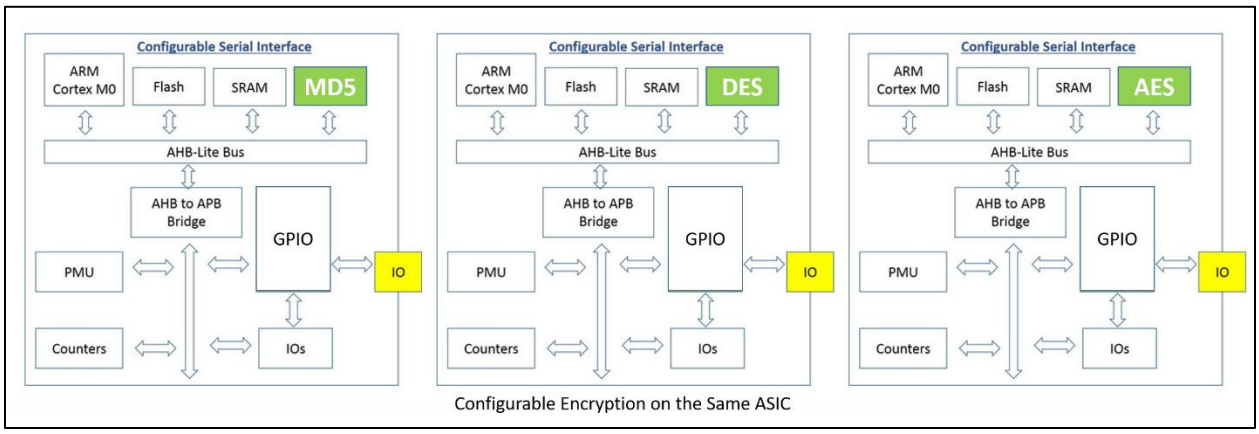

Figure 7: Configurable Encryption - An eFPGA allows a design to include different encryption engines for different end products, expanding its functional use for different security realms

For example, in cases of highly sensitive bitstream content, the system architect can implement military-grade encryption and omit commercial-grade protections, making the chip design more efficient. This customized security approach offers substantial protection from reverse engineering processes developed by those nation-states invested in attacking traditional commercial FPGA products. Additionally, in the scenarios where an eFPGA is replacing an onboard FPGA, all the board level traces that were vulnerable to being probed would be included on the ASIC where these communication lines can no longer be sniffed for information as illustrated in Figure 8.

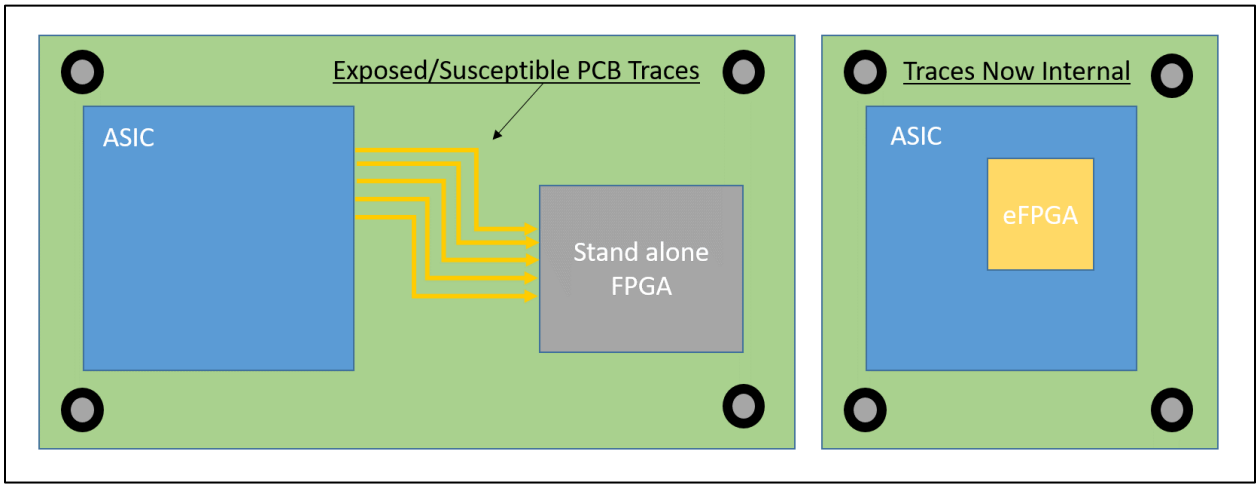

Figure 8: Exposed Traces - the communication traces on a printed circuit board (PCB) between an ASIC and FPGA parts represents a security vulnerability. Moving the FPGA fabric on the ASIC by means of an eFPGA fabric eliminates this weakness

3.6. Securing sensitive algorithms from the ASIC manufacturer

Integrating an eFPGA into an ASIC design can help improve supply chain security by enabling greater control of a program's hardware assurance level and reducing the potential exposure of design details to a third party. The eFPGA design can be ported to, and manufactured in a trusted facility providing confidence in the fabrication process and the manufactured end products. If a Trust certified foundry is not available, a program with a sensitive design could have the product fabricated at an unclassified facility and then program the sensitive algorithm into the eFPGA after delivery of the manufactured part. This could allow for the use of commercial foundries while protecting against adversarial compromise of sensitive algorithms.

3.7 Tradeoffs Summary

The following table summarizes the tradeoffs of using an eFPGA in a DoD ASIC based system.

Table 1: Benefits and tradeoffs of using eFPGA

Benefits of using eFPGA	Tradeoffs of using eFPGA
Allows a program to mitigate risks involved with immature algorithms	Adds complexity to the physical integration of the ASIC design

Benefits of using eFPGA	Tradeoffs of using eFPGA
Allows a program to utilize the same silicon chip for multiple revisions of the product in support of its lifecycle management	The models may not fully support all the needed process requirements
Can be used to improve SWaP in multichip systems	The security features that come bundled with the commercial FPGA need to be developed for the eFPGA
Lowers the cost in higher volume products as compared to using an FPGA	Can be more expensive for very low volume programs
Includes some inherent security enhancing characteristics	
Can be used for algorithm redaction when fabricating a DoD system	

4. NSA evaluation of eFPGAs

Since the current generation of eFPGAs is used by various U.S. Government agencies, labs, and DoD suppliers, the National Security Agency's Hardware Solutions Division evaluated the efficacy of incorporating this feature into DoD systems. This research effort included five phases, identified as Rocky I through Rocky V. This section provides information and observations stemming from that effort.

4.1. Rocky I

The Rocky I report, "Rocky Phase 1 eFPGA Overview," chronicles NSA's basic research on eFPGA vendors, their products, their business models, and their design flows. This non-proprietary report also identifies the advantages of the using this IP type and the applications where eFPGAs can have the most impact.

4.2. Rocky II

The vendor-proprietary Rocky II report, "Rocky Phase II eFPGA Deliverables," details NSA's evaluation of vendors' software and tutorials to design and create a custom-built eFPGA block for evaluating their deliverables, documents, and support. It also assesses the quality of the tools used to perform this task and the deliverables provided to integrate the IP into an ASIC.

4.3. Rocky III

The Rocky III report, “Rocky Phase III eFPGA Configuration,” is also vendor proprietary. It describes use of a vendor development board or test hardware to evaluate back-end software support for developing and loading applications into the eFPGA that is incorporated into the ASIC hardware. This evaluation included implementing a custom design, evaluating a design, creating bitstreams, and summarizing the results of the effort.

4.4. Rocky IV

The Rocky IV report, “Rocky Phase IV eFPGA Security and Assurance,” summarizes the security related benefits and vulnerabilities of using an eFPGA, as well as the recommended security architectures for protecting sensitive data associated with these blocks. This report is non-proprietary.

4.5. Rocky V

The Rocky V report, “Rocky Phase V eFPGA ASIC Demonstration,” describes using a design vehicle to incorporate an eFPGA block from one of the vendors into an ASIC, manufacturing it in a Trusted Foundry, and evaluating the resulting parts. This report is vendor proprietary.

These reports can be requested by sending an email to JFAC_HwA@radium.ncsc.mil.

5. Study observations

The five phases of evaluation described in section 4 yielded a great deal of information regarding the strengths and weaknesses of eFPGAs. In addition to the comparisons with commercial FPGAs, the studies illustrated that eFPGA macros are a unique and powerful ASIC IP function. Many of the NSA JFAC HwA evaluation team’s observations are captured in sub-sections 5.1 through 5.7. These observations are categorized as eFPGA vendor information, evaluated eFPGA products, supporting software, ASIC deliverables, IP integration, eFPGA programming, and assurance and security.

The evaluation effort began in 2020 and the following observations and recommendations reflect the state of eFPGA offerings available in the commercial market at that time. These products may have matured and been enhanced in the intervening time. The reader is encouraged to educate themselves on any relevant updates.

5.1. eFPGA vendors

Four commercial vendors participated in this evaluation – Achronix, Flex Logix, Menta, and QuickLogic. Examining these companies, their products, and their business models generated the following observations:

- All vendors have been designing and distributing eFPGAs for at least five years and collectively supported more than 100 ASIC development efforts.
- Some of them have performed extensive work with the DoD.
- All vendors had clear business, licensing, and pricing practices. They all had worked with one or more USG programs and understood the specialized requirements that are typical of USG contracts.
- All vendors provided detailed technical support when needed.
- All vendors provided the full scope of software/tools, training, documentation, and deliverables necessary for a successful ASIC development effort.
- Three vendors are headquartered in the US. Menta, is headquartered in France, but has U.S. sales offices that can work with source restricted programs.
- One vendor can support classified and ITAR work.

5.2. Evaluated eFPGA products

The companies provided documentation and training on the architecture, features, unique capabilities, and limits of their respective products. Evaluating this information yielded the following general observations:

- All eFPGA products evaluated in this study consisted of a base programmable logic element that could be tiled into larger configurable macros. Each programmable logic element consisted of a lookup-table and a registered output. These logic elements could be tiled to create an eFPGA of a size specified by the user.
- All products were customizable to the user's needed size and technology process(es).
- All were portable to different manufacturing processes within 3-12 months.

- Each vendor product was capable of supporting at least one other type of function to be embedded in the eFPGA fabric, such as digital signal processing (DSP) blocks, memory, or a user defined custom function.
- While each vendor supported multiple foundries and processes, no single vendor product was available across all foundries and processes. However, a user should have little problem identifying a solution using their preferred manufacturing process among the group of vendors.
- All of the vendor eFPGA products were accompanied by a configuration logic module that enables user programming of the application into the device macro.
- None of the vendors provided built-in security features, such as bitstream authentication or encryption that would normally accompany FPGA devices. However, these vendors can either provide third-party solutions or guide the user to third-party vendors for assistance with security features.
- None of the vendors required user design information to build a custom eFPGA block.
- The eFPGA deliverables were sent by protected https or sftp download sites. However, none of the files were encrypted and login information was sent in the clear by email.
- Menta provided deliverables through a US-based sales office for acquisition-restricted programs.
- The vendors used architectures that utilized standard cell libraries to varying degrees. At the extremes, one vendor built their architecture solely out of standard cells, while another primarily used custom blocks. In the middle, two vendors used a combination of both. These approaches allow for quick stand up time and would allow for mixed library cell types that could target certain areas of the eFPGA for high performance cells and others for low power.
- Three vendors used SRAM bit-cells or an SRAM bit-cell/latch combination for the configuration memory. The fourth vendor used registers to store their configuration data.
- All vendors chose to create custom routing cells to optimize the area and capability of the switching resources.

In summary, there is a wide variety of products to choose from, allowing the prospective user options that best match their program needs.

5.3. Software

As with commercial FPGAs, an eFPGA requires software to synthesize, place, route, and verify the user application. Each vendor provided some tools to perform the entire eFPGA implementation flow through bitstream generation. Post bitstream processing steps, such as encryption or authentication, were not supported by any of the vendors. What follows are some general observations regarding the vendor software:

- While features and ease of use of the implementation software differed significantly among the vendors, they all provided what was needed to perform design synthesis, place and route, and configuration data generation for use in an ASIC.
- Two companies relied upon Synopsys Synplify for the synthesis portion of their implementation flow. In one case, the user had to acquire the license, and in the other, it was included in the licensing agreement.

5.4. Deliverables

An ASIC IP block, a process, and tool-specific EDA models are needed to integrate an eFPGA into a custom ASIC design. Each vendor provided a specific and limited set of deliverables to the user for their ASIC design flow. What follows are general observations regarding the deliverable files available at the time of this study:

- Each vendor provided models to support simulation, synthesis, pre/post route timing analysis, place and route, and physical verification.
- There was not significant support for power analysis in the models.
- All vendors provide manufacturing tests and support with their offerings.
- The deliverables from each vendor were checked for pin matching, correct pin direction, compliance with the format standard, and for functional correctness. Multiple files from two vendors had to be corrected.
- As mentioned previously, standard cells are used in the layout of the different eFPGA architectures to one degree or another. These are hardened into the final embedded macro. This approach can create complications for generating and

applying back-annotated timing in the form of an SDF file. Two vendors provided SDF and a workflow to apply the timing to the macro for simulation. Two vendors relied only on their simulation models to provide timing. This created functional verification challenges at the lower geometries where accurate timing is of great importance. Here the vendors would emphasize the use of Liberty files to validate timing and simulation to verify function separate from accurate timing. All four vendors provide a means to overcome this timing challenge. However, of all the areas of ASIC development, analyzing post-route timing was the most challenging.

- The quality of the provided simulation and timing analysis models varied widely. Two products did not support dynamic loading on the outputs of the eFPGA macro. This would result in the same simulated delay regardless of the loading.
- The run times of the simulation models also differed significantly. One vendor's model significantly increased the simulation run time because of its design and incompatibility with Cadence multi-threaded simulator. The team could not determine if the simulator or the model was the cause. The model did simulate correctly using a single-thread license, but run time was very long.

5.5. Integration

The eFPGA has to be integrated into the ASIC with great care. The eFPGA has hundreds to thousands of pins on its periphery that have to synchronize with the ASIC design logic and the eFPGA internal application. Ensuring that the block is inserted and correctly attached to the ASIC logically and electrically requires precision and is time consuming. Other observations include the following:

- Integration of the eFPGA macro was straight forward and in a manner similar to that of any other embedded macro except with a greater pin count.
- The high number of manufacturing test pins did create some difficulty integrating the macro.

5.6. Programming

The user application needs to be programmed into the eFPGA as a part of simulation verification and into hardware after ASIC manufacturing. Some general observations regarding that process include the following:

- All vendor products came with an HDL configuration unit that was either embedded in the eFPGA macro or synthesized into the ASIC logic.
- The programming software worked like any FPGA implementation tool that synthesized code, implemented the application, and then generated a configuration file.
- The vendors did not provide encryption or authentication support with their products. It was up to the developer to build that into their ASIC.
- All products offered serial and parallel programing support.

5.7. Assurance and security

The NSA JFAC HwA evaluation team made the following observations about assurance and security:

- The eFPGA has no accompanying security features. Any requirements for protecting the confidentiality of the application design fall to the user.
- From an assurance standpoint, eFPGA should be treated as an ASIC third-party IP and subjected to the requisite level of acceptance testing recommended by the JFAC ASIC Assurance Best Practice Guides.
- Vendor software should undergo all the recommended acceptance testing contained in the JFAC ASIC Assurance Best Practice Guides.
- At the time of the evaluation, one of the vendor software suites had a “severe” level rating in the MITRE Common Vulnerabilities and Exposures (CVE) database.
- Few vendors used secure communications or data delivery methods involving digital signatures or encryption.

6. Recommendations to USG

After completing the five Rocky phases, the NSA JFAC evaluation team compiled the following list of recommendations for the USG regarding the use of eFPGAs in DoD systems:

- The eFPGA should be viewed primarily as an ASIC IP block and not as an FPGA substitute.

- The USG should engage with eFPGA vendors to communicate USG assurance and security needs and to influence product design and development in a manner that would benefit DoD programs.
- The USG should recognize eFPGA as a means to maintain confidentiality of sensitive portions of DoD-specific ASIC designs during manufacture. This fact makes the continuing maturity of this product a DoD ASIC assurance priority.
- Since eFPGA macros are custom IP blocks, they represent a target of opportunity for an adversary to attack specific programs – their targetability is built in. Accordingly, DoD should engage JFAC to develop assurance best practice guidance to support this product.
- The USG should investigate the development of security functions and architectures to protect these devices and their bitstreams from malicious attacks. This could include government-off-the-shelf encryption and authentication engines to protect the programmable portion of the macros.

7. Recommendations to users

For users of eFPGA macros in their ASIC designs, the NSA JFAC HwA evaluation team makes the following recommendations in the following sub-sections 7.1 through 7.4):

7.1. Select the appropriate product

When performing the cost-benefit analysis of using eFPGA in an ASIC, consider the following:

- eFPGA macros are large compared to the ASIC logic it is intended to implement. The program should ensure that including the eFPGA does not exceed the physical space constraints. For example, if including the eFPGA changes the planned die size of 3mm x 3mm to 5mm x 5mm, does this push the program out of budget? If so, then plan for the larger size.
- Which vendor product already supports the foundry and process geometry being targeted by the program? If the eFPGA needs to be ported to a new process technology, the program will need to understand the additional non-recurring engineering (NRE) costs, the schedule cost to perform the work, and the cost to qualify the product in that new technology. Multi-project wafer runs can be used to save money when qualifying a new product if available.

- Ensure the product supports the number of separate clock signals/domains needed.
- Ensure the product supports the types of global resets needed; synchronous, asynchronous.
- Understand the number of top-level test pins needed to support the manufacturing tests.
- Understand the fault coverage provided by the vendor test methodology.
- Understand the number of top-level configuration pins needed to program the fabric.
- If additional functional blocks are embedded in the eFPGA, ensure the schedule, the space, and the NRE costs are understood. This includes memory, DSP, and custom functions.
- Understand the costs of using multiple instances or multiple sizes of eFPGA macros in the ASIC.
- The user should determine early in the design phase if the estimated bitstream size is supportable by the system.

7.2. Develop the custom macro

When developing the application and determining the correct size of the eFPGA, the user should consider the following:

- Do not provide more information to the vendor than is needed. This IP block is a custom function that provides an inherently targetable attack surface to an adversary. The vendors do not require any ASIC design information to develop and provide the IP.
- Work with the vendor to obtain placement models for their tools to determine if the planned application will fit within the estimated size of the device. Leave area margin for future revisions to the planned eFPGA macro.
- Consider the eFPGA profile before defining the final macro. Designs with high I/O count and short pipelines might benefit from tall-narrow designs to minimize distance from inputs (left side) to outputs (right side). Designs with low I/O count and long pipelines could benefit from short-wide designs.

- Consider the routing obstacles that embedded macros, such as memories or DSPs, represent when embedded in the fabric. This can impact the best I/O placement.
- Designs that require sleep or power down modes can be significantly larger, more complex, and more difficult to verify. Ensure these types of requirements are fully examined prior to committing to these features.
- Examine the test requirements for embedded macros in the eFPGA, such as memory and DSP. These can represent substantial efforts, test complexity, and additional routing congestion.
- Determine the configuration load time and confirm it is within the budgeted start up time for the ASIC.
- Clearly communicate the exact process, voltage, and temperature corners that are needed from the vendor to perform timing and power analysis.
- In the case of large macros, how can voltage droop be calculated?

7.3. Ensure acceptance testing for all eFPGA deliverables

1. The user should insist that deliverables be provided in digitally signed packages and encrypted. These should not be sent by email in the clear without encryption.
2. Each delivery should be identifiable by a revision number.
3. The user should save the deliverables in a revision control system.
4. The program should plan to perform acceptance testing on the electronic design automation files provided by the vendor. This testing should:
    a. Confirm that all of the expected models were provided and that they contain the expected support need for the user's tool flow. Of particular concern are:
        i. Simulation models – they should provide the ability to be run "as configured" or with configuration sequence. It should be understood how timing will be modeled in each stage of the design process.
    b. Confirm that pin name and pin direction match across models.
    c. Confirm that each model can be ingested by its target tool set.

d. Verify that the Liberty models can support the needed timing analysis, simulation, and power analysis.

e. Verify that physical verification can be performed using the provided layout and schematic models. The NSA JFAC HwA evaluation team recommends avoiding black box GDSII/OASIS deliveries that are back filled at the manufacturer especially for advanced technology nodes. The black box process can complicate double patterning processes and open physical verification to hidden violations.

f. Verify that the manufacturing tests can be run in simulation and used at wafer testing.

g. Verify that the documentation is complete and provides directions regarding integration of the macro.

7.4. Design integration

When beginning the ASIC development flow, the user should:

1. Identify and understand any limitations of the models with respect to timing analysis and back-annotated timing.
2. Consider that the I/O location assignment is extremely important when using eFPGA. They have 100s to 1,000s of I/Os that can create routing bottlenecks and impact timing.
3. Develop a plan to verify all of the connectivity into and out of the macro.
4. Work with the vendor to determine a simulation methodology to exercise the loading of the target application into the eFPGA simulation model and for simulating the ASIC with the macro already configured. Determine how to verify a correct load of the bitstream.
5. Avoid mixed languages (Verilog vs VHDL) at the eFPGA-ASIC interface. It has the potential to create simulation mismatches between the register-transfer level and netlist simulations.
6. Expect additional effort to create clock trees that are balanced with the eFPGA internal endpoints. Special routing constructs may be necessary.

8. Conclusions

With DoD raising the importance of custom microelectronic hardware assurance, the eFPGA stands as an important means to support design confidentiality in the manufacturing process. This, combined with its other benefits, raise the value of this product to DoD programs.

Following the Rocky evaluations, the NSA's JFAC Hardware Assurance team deems this IP ready for use by ASIC designs. Each of the evaluated eFPGAs had shortcomings that can complicate the ASIC development phase, but none of them were serious enough to prevent a successful design completion. If an end user gives attention to the recommendations contained in this summary document, they will have the information needed to ensure a successful ASIC effort.

Finally, if a program has questions regarding this report or requires assistance, please contact NSA JFAC at JFAC_HwA@radium.ncsc.mil.