

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ADVANCED MICRO DEVICES, INC. and PENSANDO SYSTEMS, INC.,

Petitioners,

v.

CONCURRENT VENTURES, LLC and XTREAMEDGE, INC.,

Patent Owners.

Case IPR2025-00223

U.S. Patent 10,985,943

**DECLARATION OF DR. PAUL S. MIN IN SUPPORT OF
PATENT OWNERS' RESPONSE**

TABLE OF CONTENTS

I.	SCOPE AND SUMMARY OF OPINIONS.....	5
II.	MATERIALS CONSIDERED	6
III.	QUALIFICATIONS AND PROFESSIONAL BACKGROUND	7
IV.	UNDERSTANDING OF LEGAL PRINCIPLES	11
	A. Obviousness.....	12
	B. Claim Construction	14
V.	PRIORITY DATE OF THE '943 PATENT	14
VI.	LEVEL OF ORDINARY SKILL	15
VII.	TECHNICAL BACKGROUND	16
VIII.	SUMMARY OF THE '943 PATENT	18
	A. Overview of the Patent	18
	B. District Court Allegations	22
IX.	CLAIM CONSTRUCTION	22
	A. “programmable logic device”	22
	B. “hardware-based” “router with a port”	23
X.	THE ASSERTED PRIOR ART	24
	A. U.S. Patent 10,776,142 (“Seshadri”) (EX1004).....	24
	B. U.S. Patent Application Publication No. 2018/0191642A1 (“Biederman”) (EX1005)	26
	C. U.S. Patent Application Publication No. 2019/0095113A1 (“Atsatt-113”) (EX1006).....	26
	D. “Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center” (“Tarafdar”) (EX1007)	29

E.	U.S. Patent No. 8,015,386 (“Kulkarni”) (EX1008)	30
F.	U.S. Patent Application Publication No. 2019/0041923A1 (“Atsatt-923”) (EX1009).....	30
XI.	PETITIONERS AND DR. ALPERT DO NOT SHOW THAT ANY CHALLENGED CLAIM WOULD HAVE BEEN OBVIOUS.....	30
A.	Petitioners Fail to Prove their Combination of Seshadri and Biederman (Grounds 1 and 2) Disclosed or Renders Obvious Claims 1–5, 9–14, and 18.....	30
1.	Claims 1 and 10.....	30
a.	Seshadri Does Not Disclose or Render Obvious that the Alleged “Bridge” is “Lockable to prevent user access,” and “Unlockable to enable user access” “to each of the sandboxes and the user- definable programmable electronic circuits therein.”	31
b.	Seshadri Does Not Disclose or Render Obvious that the Alleged “Router with a Port” are “Hardware-Based.”	39
2.	Claims 5 and 14.....	46
a.	Seshadri Does Not Disclose or Render Obvious that the Claimed “Bridge” with a “Security Layer.”	46
B.	Petitioners Fail to Prove their Combination of Atsatt-113 and Biederman (Grounds 3 and 4) Disclosed or Renders Obvious Claims 1–5, 9–14, and 18.....	47
1.	Claims 1 and 10.....	48
a.	Atsatt-113 Does Not Disclose or Render Obvious that the Alleged “Bridge” is “Lockable to prevent user access,” and “Unlockable to enable user access” “to each of the sandboxes and the user- definable programmable electronic circuits therein.”	48

(1)	Atsatt-113’s Masks Do Not “Prevent User Access” Or “Enable User Access” “to each of the sandboxes.”	49
(2)	Atsatt-113’s “Interconnection Resources” Cannot Be Locked Or Unlocked And Do Not Enforce The “Masks.”	52
b.	Atsatt-113 Does Not Disclose or Render Obvious the Claimed “Router with a Port” In The Alleged “First Region.”	56
c.	Atsatt-113 Does Not Disclose or Render Obvious that the Alleged “Router With a Port” are “Hardware-Based.”	60
2.	Claims 4 and 13.....	61
3.	Claims 5 and 14.....	64
a.	Atsatt-113 Does Not Disclose or Render Obvious that the Claimed “Bridge” with a “Security Layer.”	64
C.	Ground 5 Does Not Address The Problems In Grounds 1–4.....	65
D.	Ground 6 Does Not Address The Problems In Grounds 1–4.....	65
E.	Ground 7 Does Not Address The Problems In Grounds 1–4.....	65
XII.	SIGNATURE.....	67

I. SCOPE AND SUMMARY OF OPINIONS

1. I, Dr. Paul S. Min, am making this declaration in support of Concurrent Ventures, LLC and XstreamEdge, Inc.’s Patent Owners’ Response in the matter of IPR2025-004223 before the Patent Trial and Appeal Board (PTAB) of the United States Patent and Trademark Office, which involves the Petition for Inter Partes Review (“IPR”) of U.S. Patent No. 10,985,943 (the “’943 patent”) filed by Advanced Micro Devices, Inc. and Pensando Systems, Inc. (“Petitioners”).

2. As explained in detail below, I am qualified to offer opinions from the perspective of a person of ordinary skill in the art (“POSITA”) at the time of the invention and can render an objective opinion regarding the issues raised in this IPR.

3. I have been asked to analyze the Petition (Paper 1) and Dr. Alpert’s declaration (EX1003) in support thereof, as well as the ’943 patent and its file history, and all identified prior art. I have been asked to provide my opinions about certain obviousness theories and opinions in Dr. Alpert’s declaration. I have also been asked to provide opinions about how a POSITA would have understood and interpreted the ’943 patent, including its claims; how a POSITA would have understood and interpreted the Petition’s alleged prior art references; and a POSITA’s knowledge of the state of the art as it relates to addressing these obviousness theories and opinions. My opinions and conclusions are set forth in this

Declaration. I understand that this Declaration is being submitted in connection with Patent Owners' Response to the Petition.

4. For purposes of this Declaration, I was asked to consider claims 1–18 of the '943 patent, which I understand are the challenged claims. *See* Pet., 16. I understand that claims 1 and 10 are the independent claims.

5. This Declaration and its exhibits, in addition to my education, training, and experience in technology fields pertinent to the '943 patent, form the basis of my opinion.

6. I am not offering any conclusions as to the ultimate determinations that I understand the Patent Trial and Appeal Board ("Board") will make in this proceeding. I am solely providing my opinions on the technical interpretations, characterizations, theories, and opinions that I understand are involved in this proceeding.

7. I may offer additional opinions based on my further review of the materials in this case, including opinions and/or testimony of other expert declarants.

II. MATERIALS CONSIDERED

8. In forming the opinions in this Declaration, I reviewed the Petition (Paper 1), Dr. Alpert's declaration (EX1003), the materials considered by Dr. Alpert (EX1003, Appx. 2), the prior art and supporting materials cited by Petitioners, the

'943 patent, and the '943 patent's file history. I have also cited various other materials throughout my declaration that I considered as part of my analysis.

III. QUALIFICATIONS AND PROFESSIONAL BACKGROUND

9. I believe that I am well qualified to serve as a technical expert in this matter based upon my educational and work experience. My curriculum vitae ("CV") is submitted as Exhibit 2010.

10. I received a B.S. degree in Electrical Engineering in 1982, an M.S. degree in Electrical Engineering in 1984, and a Ph.D. degree in Electrical Engineering in 1987 from the University of Michigan in Ann Arbor. I received several academic honors, including my B.S. degree with honors, a best graduate student award and a best teaching assistant award during my M.S. study, and a best paper award from a major international conference for reporting results from my Ph.D. thesis.

11. After receiving my Ph.D., I worked at Bellcore in New Jersey from August 1987 until August 1990. At Bellcore, I was responsible for evolving the public switched telephone network (POTS) into a multi-services voice and data network that incorporated packet switches, optical technologies, and wireless technologies.

12. In September 1990, I joined the faculty at Washington University in St. Louis. In July 1996, I was promoted to an Associate Professor of Electrical

Engineering with tenure. I am currently a Senior Professor at Washington University of Electrical and Systems Engineering. I have also served as the Chair of the Graduate Curriculum (2000-2002) and the Chair of the Undergraduate Curriculum (2011-2014) for the Electrical and Systems Engineering department.

13. At Washington University, I have conducted research in communication, computing, and related electronic hardware and software. My research group has pioneered a new paradigm for designing electronic circuits that can alleviate the speed and performance mismatch against optical technology. I have received several grants from the U.S. Federal Agencies, including the National Science Foundation and the Defense Advanced Research Project Agency, and numerous contracts from companies and organizations around the world. As part of my research activities, I have designed and implemented a number of System-on-Chip (SoC) devices, which incorporated multiple processor-cores that communicate and cooperate with others.

14. As a faculty member at Washington University, I have taught a number of courses in electronics, communication, and computing at both the undergraduate and graduate levels. For example, I have taught communication theory (Washington University ESE 471), transmission and multiplexing (Washington University ESE 571), and signaling and control of communication networks (Washington University ESE 572).

15. I have supervised numerous graduate students, many of whom received a doctoral degree under my guidance. A number of doctoral theses that I have supervised relate specifically to the computing and communication technologies related to this proceeding. My students and I have published a number of peer-reviewed articles in these topics.

16. In addition to my responsibilities as a university faculty member, I have founded two companies. In May 1997, I founded MinMax Technologies, Inc., a fabless semiconductor company that developed switch fabric integrated circuit chips for the Internet. In March 1999, I founded Erlang Technology, Inc., a fabless semiconductor company that focused on the design and development of integrated circuit chips and software for the Internet. One of Erlang's products received a best product of the year award in 2004 from a major trade journal for the electronics industry. As part of my company activities, I have prototyped numerous chassis systems, which included multiple modular cards performing various tasks (e.g., switching card, memory card, input/output card, power supply card, etc.). In doing so, I became intimately familiar with various module standards such as CompactPCI and ATCA (Advanced Telecom Computing Architecture).

17. I developed and implemented a number of semiconductor integrated circuit chips. These integrated circuit chips include switch fabrics, network processors, protocol processor high speed interfaces, protocol transposers,

controllers, content search engines, look up table processors, and memory cells and arrays.

18. I am a named inventor on eleven U.S. patents, many of which are directly related to computing and communication technologies. I have extensively published technical papers in international conferences and journals, technical memoranda, and reports, and given a number of seminars and invited talks. I have organized several international conferences and served as an international journal editor.

19. I am a member of and have been actively involved in a number of professional organizations. For example, I have served as the Chair of the Saint Louis Section of the IEEE with more than 3,000 members (2014), and a member of the Eta Kappa Nu Honor Society for electrical engineers. I have also been an Ambassador of the McDonnell International Scholars Academy (2007-2013).

20. I was a member of the Presidential Business Advisors Committee (to President George W. Bush) and was named 2002 Businessman of the Year by the Wall Street Journal for the State of Missouri, for my entrepreneurial efforts.

21. I have received multiple awards for my publications and contributions to technical communication systems. In 1988, I received the Best Paper Award at 18th ISATA Award of Technical Excellence in Florence, Italy. I was also a Rockwell Fellow at the University of Michigan in Ann Arbor in 1988 and 1989. I received a

Research Initiation Award in 1993 from the Defense Advance Research Project Agency (DARPA). I received the Best Paper Award at MOBILITY 2011 in Barcelona, Spain. In my nearly 40 years of professional experience, I have acquired significant knowledge about telecommunications, computer, and electronics industry standards and standard setting organizations such as ANSI, IEEE, ITU, IETF, Bellcore, and 3GPP.

22. Additional details of my education and work experience, awards and honors, and publications that may be relevant to the opinions I have formed are set forth in my curriculum vitae, as well as a list of cases in which I have testified as an expert at trial or by deposition. *See* EX2010.

IV. UNDERSTANDING OF LEGAL PRINCIPLES

23. At the outset, I want to clarify that all of my analysis and opinions in this declaration are from the perspective of a POSITA at the time of invention. To the extent that portions of this declaration do not expressly include language indicating that the analysis is from the perspective of a POSITA at the time of invention, it should be understood that my analysis and opinions are from the perspective of a person of ordinary skill in the art at the time of invention. For example, my opinion that a reference does not disclose or teach a claim limitation should be understood to mean that, in my opinion, a POSITA at the time of invention would have interpreted the reference as not disclosing the limitation. Likewise, my

opinions regarding obviousness are from the perspective of a POSITA at the time of the invention.

24. I do not offer opinions of law. The opinions I am expressing in this Declaration involve the application of my knowledge and experience to the evaluation of the challenged patent and certain art with respect to the challenged. The following paragraphs express my understanding of legal principles related to my analysis.

A. Obviousness

25. I understand that a claim is unpatentable under 35 U.S.C. § 103 if the differences between the claimed subject matter and the prior art are such that the claimed subject matter as a whole would have been obvious to a person of ordinary skill in the art at the time of the invention. I also understand that an obviousness analysis is based on the scope and content of the prior art, the differences between the claimed subject matter and the prior art, the level of ordinary skill in the pertinent art at the time of the invention, and objective indicia of nonobviousness, such as commercial success, industry praise, unexpected results, copying, and other such factors.

26. It is my understanding that, to find a claimed invention obvious in light of a combination of prior-art references, a person of ordinary skill in the art at the time of the invention must have had reason to combine the teachings of the prior-art

references to achieve the claimed invention, and that the person of ordinary skill must have had a reasonable expectation of succeeding in combining the teachings to arrive at the claimed invention. I further understand that a reason to combine or expectation of success is undermined if the combination or modification of the technology disclosed in the prior art would interfere with an objective of the technology disclosed in the prior art or if the prior art teaches away from making the combination or modification. I understand that a prior art reference is said to teach away from the claimed invention if a person of ordinary skill in the art at the time of the invention, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken in the claim.

27. I understand that an assessment of what a reference discloses or teaches, for purposes of an obviousness analysis, must be conducted from the perspective of a person of ordinary skill in the art at the time of the invention. In other words, a reference discloses or teaches a claim limitation only if a person of ordinary skill in the art, at the time of the invention, would have interpreted the reference as disclosing the claim limitation. I understand that reliance on common sense, common wisdom, and common knowledge to inform an obviousness analysis must be explained with sufficient reasoning. I understand that assessment of obviousness

from the perspective of someone skilled in the art before the invention was made, without the benefit of hindsight.

28. I understand that it is not permissible to use hindsight after viewing the claimed invention to determine the question of obviousness or to rely on the teachings of the claimed invention in determining whether a POSITA would find the invention obvious.

B. Claim Construction

29. I understand that claims and their terms are construed, that is, interpreted, in accordance with their ordinary and customary meaning as understood by a person of ordinary skill in the art at the time of the invention when read in the context of the patent's specification and prosecution history. I further understand that if the patent's specification reveals a special definition given to a claim term by the inventor, then the inventor's special definition governs, even if it differs from the term's ordinary meaning.

30. I understand that there are certain terms for construction that are at issue in this proceeding. I have discussed these terms and my understanding of them further below.

V. PRIORITY DATE OF THE '943 PATENT

31. Dr. Alpert states that the time of the invention is July 15, 2018. For the purpose of my opinions in this Declaration, I have been asked to assume the '943

patent is entitled to that priority date. Therefore, when I refer to a POSITA at the time of the invention in my Declaration, I am referring to a person of ordinary skill in the art as of July 15, 2018.

VI. LEVEL OF ORDINARY SKILL

32. Dr. Alpert set forth a level of ordinary skill as follows:

I came to the conclusion that a person of ordinary skill in the field of art of the '943 patent would have been a person with a bachelor's degree in computer engineering, electrical engineering, or a similar major with at least two years of experience in the field (described above) or a person with a master's degree in computer engineering, electrical engineering, computer science, or a similar field with a specialization in the field (described above). A person with less education but more relevant practical experience may also meet this standard.

EX1003, ¶ 38. Dr. Alpert went on to list skills and experience a POSITA would have had in the 2018 timeframe. *Id.*, ¶ 39.

33. I understand that the following factors may be considered in determining the appropriate level of skill in the art: (1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field. I understand that the person of ordinary skill in the art is not an actual individual, but instead is a construct for a hypothetical person who is presumed to be aware of all the pertinent prior art. I have addressed relevant information regarding these factors and applied this understanding below.

34. As summarized above with regards to my qualifications and experience, I possessed at least the level of a POSITA at the time of the invention of the '943 patent because I had a Ph.D., M.S., and B.S. in Electrical Engineering. EX2010. I am currently a Senior Professor in the Department of Electrical Systems Engineering at Washington University, St. Louis, and have worked in the field of electrical engineering and data processing, streaming architecture, and data networks for more than thirty years. *Id.*

VII. TECHNICAL BACKGROUND

35. Data-processing platforms often employ programmable logic devices (“PLDs”) and modular components that communicate through internal messaging networks, enabling flexible data-flow architectures. In these systems, internal communication is commonly mediated by message interfaces that route data and control information among processing modules memory components. These architectures provide configurability and performance advantages and also introduce technical challenges in controlling how different modules interact with shared infrastructure and reconfigurable resources, including sandbox regions.

36. A technical challenge arises when portions of a PLD are intended to be configurable by users while other portions are to remain protected. For example, PLDs may include distinct regions for internal communication infrastructure and for user-defined circuitry, with internal routers and bridges coordinating data flow

between modules. In such systems, internal communication may include routing components, security layers, and handling logic that govern how modules exchange data and how external interfaces connect to internal resources. Because these components form the backbone of the system's data flow, uncontrolled modification of them could be dangerous.

37. To address these concerns, systems may implement mechanisms that restrict modification of certain components while permitting modification of configurable regions. In particular, a bridge or routing component may be configured so that it cannot be altered by user configuration activity, even when other portions of the device are reconfigured. This separation ensures that user-defined modules can be modified without disrupting the underlying communication infrastructure. In practice, such mechanisms operate as a form of locking that prevents changes to protected circuitry while allowing changes elsewhere, thereby enforcing structural boundaries between protected and configurable regions within the device.

38. Related to this locking concept, security layers may control whether a user is permitted to modify configurable regions of the system. These functions may include authentication or other forms of permission verification before reconfiguration is allowed. In this way, the system distinguishes between authorized and unauthorized activity and ensures that reconfiguration occurs only when appropriate conditions are satisfied.

39. These mechanisms are significant in modular platforms that support partial reconfiguration, where different portions of a PLD may be independently modified without halting system operation. The ability to selectively restrict modification of critical communication components while permitting modification of other regions is important. The technical problem addressed by these mechanisms is thus not simple reconfiguration, but the control of which portions of a complex architecture may be altered and under what conditions, while preserving the system's routing and communication functions.

VIII. SUMMARY OF THE '943 PATENT

A. Overview of the Patent

40. The inventors of the '943 patent developed various breakthroughs in dedicated hardware technology that enable data processing faster than software, but also allow the flexibility required to accommodate for rapidly changing data formatting requirements or new data processing applications. EX1001, 1:8–21. For example, video processing is one such data processing application, in which many new different standards are constantly and rapidly introduced, resulting in the need for flexible solutions for data processing. *Id.*

41. In response to these problems, the inventors of the '943 patent innovated a novel programmable device for data flow processing in a user-configurable server. *Id.*, Abstract. For instance, the programmable device comprises

a PLD that has a first region and a second region, where the first region includes a router with a port enabling communication between the router and the second region and where the second region comprises one or more sandboxes with user-definable electronic circuits. *Id.*, 1:25–49; cl. 1.

42. The router also includes a bridge that is lockable and unlockable to prevent and enable user access to each of the sandboxes and the user-definable programmable electronic circuits therein, respectively. EX1001, 1:25–49; cl. 1. In that regard, the user-definable electronic circuits in each of the sandboxes can be accessed and configured in different ways to accommodate differing data flow processing architectures, without compromising the overall security of the programmable device. *Id.*

43. For example, Figure 3 of the '943 patent (reproduced below) illustrates a programmable device for servers that is an FPGA (122) in a pod (104), *e.g.*, mounted to a printed circuit board that is one of one or more printed circuit boards in the pod (104). EX1001, 4:39–47. The programmable device could also be in a card (106). *Id.* The programmable device has a Module Message Interface (“MMI”) router (124), a bridge (128), and sandboxes (132) for configurable modules. The router can be hard-ware- or firmware-based, where hardware-based means that the router lacks a processor. *Id.*, 5:1-8.

44. The programmable device could be, for example, an FPGA (122) as depicted in Figure 3, a PLD or an ASIC, or more than one of these, or some combination of these, or other programmable device that can configure electronic circuitry, in some versions including one or more processors or controllers. *Id.*, 4:27–38.

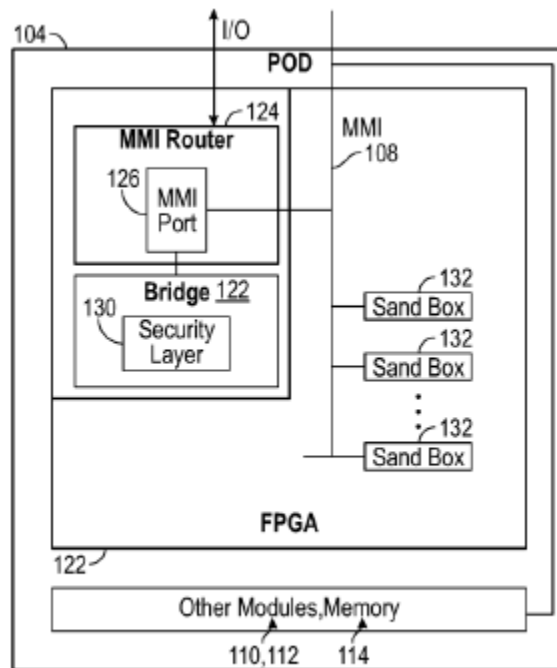


FIG. 3

45. The FPGA (122) of Figure 3 can have multiple sections, one of which can be a bridge (128) for external communication, *e.g.*, a serial connection to an external network or a parallel connection to an external bus and shown generally as Input/Output (I/O), and internal communication. *Id.*, 4:48–59. Thus, the bridge (128) may be considered a communication bridge between the outside of the server or system, and internal modules and/or memory, and has at least two ports, I/O for

external communication and MMI for internal communication, and the bridge may be a module that sits on a router port, for purposes of crossing the I/O type or protocol. *Id.*

46. As further shown in Figure 3, the bridge (128) may contain a security layer (130), and in some embodiments, a firewall for MMI message traffic based on static or dynamic policy or rules. *Id.*, 4:60–62. As such, the MMI network connection (108) connects to one or more sandboxes (132) in other sections of the FPGA (122) and can connect to other modules elsewhere in the pod (104). *Id.*, 4:62–65. One of the functions of the security layer (130) could be to protect the circuitry of the bridge, including the router (124) from being reprogrammed. *Id.*, 5:39–49. Another function of the security layer (130) may be authentication of a user or other verification of permission to reconfigure a sandbox (132). *Id.*

47. Each of the one or more sandboxes (132) is designated for programmable electronic circuits with the intent of providing user-programmable functionality for user customization of one or more pods (104) or cards (106). *Id.*, 5:19–25. For example, a user could define an analog, digital, mixed-signal hardware module (110), or a software module (112) (or more than one of these or some combinations of these) in one or more sandboxes (132). *Id.*, 5:25–34. In some embodiments, the bridge (128) is locked to prevent user access in the FPGA (122)

so that a user can configure or reconfigure one or more modules in one or more sandboxes (132) but cannot reconfigure the bridge (128). *Id.*

B. District Court Allegations

48. Dr. Alpert repeats certain infringement allegations from the related district court litigation, but he does not tether these claims to his opinion on the prior art. Based on my reading, I disagree with the limited scope of his summary, namely his summarizing hundreds of pages of infringement contentions into no more than five paragraphs.

IX. CLAIM CONSTRUCTION

49. In my opinion, a POSITA would have interpreted each claim term of the '943 patent, when read in the context of the specification and prosecution history, to have their plain and ordinary meaning, with the exception of the “hardware-based” claim term discussed below. Except for the specific instances discussed below, I have applied the plain meaning of the claims in forming my opinions.

A. “programmable logic device”

50. I understand that Petitioners have proposed a construction for “programmable logic device” in district court. I analyzed the Petitioners’ proposed construction below and disagree that a “programmable logic device” must mean “an IC where the user can alter the connection pattern of the circuits.” Pet., 15. However, for purposes of analyzing the Petition, I have applied Petitioners’ proposed

construction because Petitioners' arguments fail regardless of whether this proposed construction is adopted.

B. “hardware-based” “router with a port”

51. In my opinion, Petitioners' arguments separately ignore that the inventors acted as lexicographers in defining the meaning of “hardware-based” for purposes of the '943 patent. The term “hardware-based router” means a router that “lacks a processor.” The '943 patent expressly distinguishes between hardware-based and firmware-based implementations:

In some embodiments, the MMI router 124 *is hardware-based, that is it lacks a processor*. In other embodiments, the MMI router 124 is firmware-based, that is it has a dedicated processor with dedicated software that is not otherwise user programmable nor part of an operating system. In some embodiments, the MMI router 124 has both hardware-based and software-based implementations.

EX1001, 5:1-8 (emphasis added). This definition makes sense. In the '943 patent, “router with a port” functionality that is executed by a processor is inherently not hardware, but instead software or firmware. In contrast, “router with a port” functionality is hardware-based when it is executed directly in circuits, without a processor.

52. I believe that Petitioners appear to base their theories on a broader and unreasonable interpretation of this limitation under which a router is “hardware-based” purely because routing functionality is implemented on the programmable fabric of an FPGA. Pet., 27, 62. For example, Petitioners argue that Seshadri's “shell

logic” and Atsatt-113’s “global resources” are the claimed router even though there is no evidence that their functionality is in hardware circuits without a processor. Pet, 24, 58. Such construction is inconsistent with the definition supplied by the ’943 patent. EX1001, 5:1-8.

53. Further, I believe this broad interpretation is contrary to how a POSITA would understand “hardware-based” in light of the specification generally (and in light of common sense) as it would effectively mean that *everything* on an FPGA is hardware-based, contrary to the explicit disclosure in the ’943 patent. EX1001, 5:1-8, 6:20-27, 6:59-60, 7:15-20.

X. THE ASSERTED PRIOR ART

A. U.S. Patent 10,776,142 (“Seshadri”) (EX1004)

54. I have reviewed the Seshadri reference. I understand that Petitioners and Dr. Alpert use Seshadri as the primary reference for Grounds 1–2.

55. Seshadri discloses techniques for reconfiguring programmable hardware in a cloud-computing environment while a client virtual machine (“VM”) remains active. EX1004, Abstract. It addresses problems that arise when a computing service provider allows customers to deploy user-defined logic onto shared, reconfigurable devices such as FPGAs. *Id.*, 1:7–31. Based on my experience, I understand that these devices, including FPGAs, contain both processors that can execute software and configurable circuits on which hardware logic can be executed.

56. In these environments, a client's custom logic sits alongside provider-controlled management logic on the same device. EX1004, 1:7–31. Seshadri explains that permitting low-level access to programmable circuits can introduce security, stability, and performance risks, particularly when the programmable logic must be updated or replaced while other users and virtual machines continue to operate. *Id.*

57. To manage these concerns, Seshadri describes an architecture built around a “shell logic” framework that encapsulates and controls the configurable circuits. EX1004, 1:64–2:12. The shell logic is divided into two primary components: a static shell and a reconfigurable shell. *Id.* The static shell contains logic that remains constant across different users and is responsible for interfacing with the host system, handling bus protocols, and enforcing security boundaries. *Id.* 2:12–25. The reconfigurable shell, by contrast, contains logic that may change when a customer loads a new hardware design. *Id.*

58. A key feature of Seshadri is that it allows the system to function during a reconfiguration event. EX1004, 13:19–39. When a client requests that the programmable circuit be updated, the static shell may temporarily disable direct communication between the host and reconfigurable circuits. *Id.* Importantly, access to the client configurable circuits is never prevented—they are always accessible. The shell only enables access to the client configurable circuits and determines what

other portions of the system the client configurable circuits can access. It does not determine whether the client-configurable circuits can be accessed.

59. Overall, Seshadri discloses a system architecture that allows a cloud provider to offer client-programmable hardware accelerators while maintaining operational safety and availability. EX1004, 3:41–64. By wrapping customer logic within a provider-controlled static shell, isolating the reconfigurable region during updates, and continuing service transactions while reconfiguration occurs, the patent enables seamless hardware modification without requiring virtual machines to be stopped. *Id.* The techniques focus on virtualization and control of programmable circuits rather than on authentication, locking, or routing.

B. U.S. Patent Application Publication No. 2018/0191642A1 (“Biederman”) (EX1005)

60. Biederman is directed to processing data streams, distributing packets among the partitions of and FPGA and dynamically resizing partitions and choosing “program to load into” the partitions. EX1005, ¶ 134, Abs., ¶¶ 27–30, Figs. 10, 8. I understand that Petitioners use Biederman to supply the “plurality of sandboxes” limitation in Grounds 2 and 4. Pet., 33, 68.

C. U.S. Patent Application Publication No. 2019/0095113A1 (“Atsatt-113”) (EX1006)

61. I have reviewed the Atsatt-113 reference. I understand that Petitioners and Dr. Alpert use Atsatt-113 as the primary reference for Grounds 3–4

62. Atsatt-113 describes techniques for maintaining partitions in a programmable integrated circuit. EX1006, Abstract. It discloses mechanisms that allow portions of the device to be updated or reprogrammed without disturbing other portions that are already configured and operating. *Id.*, ¶ 42. To accomplish this, the system introduces the concept of applying a “mask” to configuration data so that only selected resources within a defined partition are affected during a partial reconfiguration. *Id.* Atsatt-113 explains that configuration data may be compressed, decrypted, and authenticated before being written into configuration memory, and that a corresponding mask is processed in parallel so that the configuration update can be restricted to authorized regions of the device. *Id.*, ¶ 59.

63. Central to Atsatt-113 is the architecture used to isolate reconfigurable regions. Atsatt-113 contains a device controller 60 which orchestrates the device but also contains a plurality of sectors that contain a sector controller, data register, CRAM, and ERAM. EX1006, ¶ 31, FIG. 9. Sectors may either entirely or partially encompass a partition, as partitions may extend beyond sectors. *Id.*, ¶ 40. The system receives configuration data and a mask from a configuration source. *Id.*, FIG. 16. It then applies the mask to the configuration data prior to writing the data into the reconfigurable region. *Id.*, ¶¶ 47-52. By combining the mask with the configuration data, the device controller ensures that write operations modify only those bits that correspond to resources assigned to the requesting partition. *Id.* Resources that fall

outside the partition remain unchanged, even if the incoming configuration bitstream attempts to target them. *Id.* This allows partitions to exist on the same programmable device while being updated at different times. *Id.*

64. Atsatt-113 makes clear that its masking mechanism is not equivalent to a “lock” on sandboxes. The mask does not prevent user access to partitions (the claimed sandboxes) or to any particular memory regions. EX1006, ¶¶ 47-52. Instead, it functions as a filtering layer during reconfiguration. *Id.* Every partial reconfiguration transaction combines the configuration data with the mask to generate a new configuration image. *Id.* The mask defines which bits are permitted to change during that transaction. *Id.* There is no concept of acquiring or releasing a lock, and no persistent protection state associated with a region. *Id.* Protection is instead achieved by modifying the data being written so that unauthorized portions are left untouched. *Id.*, cl. 8.

65. Atsatt-113 also addresses how partitions interact with other resources on the device. Each programmable sector of the FPGA contains local configuration memory and logic resources, while certain global resources are shared across these sectors. EX1006, ¶¶ 30-31. When a partition is updated, its own allowed mask is combined with the inverted masks of other partitions to ensure that only its assigned resources are affected. *Id.*, ¶ 42. In addition, the system employs authentication keys to verify the integrity and authorization of each component before any

reconfiguration occurs. *Id.*, ¶¶ 94-96. The device controller authenticates the configuration data using a first key and independently authenticates the mask using a second key, allowing a mask to be validated and trusted. *Id.*

66. Atsatt-113 also teaches resources accessible by either the device controller or sectors. Interconnection resources are resources in the FPGA that “may be used to route signals, such as clock or data signals” (*i.e.*, not like a traditional network) whereas global resources are shared exclusively by the partitions inside the sectors. EX1006, ¶¶ 30, 55 (“For example, global resources may include input / output ports (e.g., 1048), interfaces (e.g., high-speed serial interface—HSSI), hard processor systems (HPS), a clock network, and/or other resources shared between the partitions.”). When a partition is reconfigured, the process does not “route” data packets through the global resources, nor do these resources perform switching or forwarding or any of the operational behaviors associated with a router. *Id.*

D. “Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center” (“Tarafdar”) (EX1007)

67. Tarafdar teaches the use of management/performance-analytics tools to orchestrate VMs on FPGAs in data centers. EX1007, 000002, Abstract, 000007-9. Petitioners use Tarafdar to supply the limitations of claims 6 and 15, where the router is configured to perform analytics on resources of the programmable device, on a per module basis, including determining communication bandwidth and resource utilization. Pet., 80.

E. U.S. Patent No. 8,015,386 (“Kulkarni”) (EX1008)

68. Kulkarni is directed to a “configurable memory manager” with port interfaces that are configurable for data-width. EX1008, Abstract, 2:25-37, 1:12-54. Petitioners use Kulkarni to supply the limitations of claims 7, and 16, where the one or more sandboxes each comprise port interfaces that are configurable as to data width or speed. Pet. 87.

F. U.S. Patent Application Publication No. 2019/0041923A1 (“Atsatt-923”) (EX1009)

69. Atsatt-923 is directed to PLDs that use voltage regulators for “dynamic voltage frequency scaling” to manage power consumption and thermals for groups of logic sectors. EX 1009, ¶ 1, ¶¶ 32-33, ¶ 18-24, Fig. 1. Petitioners use Atsatt-923 to supply the limitations of claims 8 and 17, where the one or more sandboxes each comprise clock cycle utilization regulation for controlling power consumption or thermal characteristics. Pet., 83.

XI. PETITIONERS AND DR. ALPERT DO NOT SHOW THAT ANY CHALLENGED CLAIM WOULD HAVE BEEN OBVIOUS

A. Petitioners Fail to Prove their Combination of Seshadri and Biederman (Grounds 1 and 2) Disclosed or Renders Obvious Claims 1–5, 9–14, and 18.

1. Claims 1 and 10

70. In my opinion, Grounds 1 and 2 fail to disclose claim 1 and 10 for two independent reasons.

71. First, in my opinion, Seshadri does not disclose a port with a bridge that is lockable or unlockable to prevent user access to each sandbox, as required by limitations 1[d] and 10[e]. EX1001, cl. 1, 10. Petitioners at most identify that the shell logic may make it so that the client configurable circuit itself cannot access other modules within the device. There is no disclosure or suggestion that access to the client configurable circuits is ever prevented, nor that access to the client configurable circuits is prevented by locking the bridge.

72. Second, in my opinion, Seshadri does not disclose a router or port that is “hardware-based” (i.e., that the router and port “lack[] a processor”), as required by limitations 1[b] and 10[b]. EX1001, cl. 1, 10. Petitioners allege that Seshadri’s routing “functionality” is “implemented in the hardware programmable fabric of the FPGA,” but the mere fact that something is programmable in an FPGA does not mean that it is hardware-based. Pet., 27.

a. Seshadri Does Not Disclose or Render Obvious that the Alleged “Bridge” is “Lockable to prevent user access,” and “Unlockable to enable user access” “to each of the sandboxes and the user-definable programmable electronic circuits therein.”

73. In my opinion, Petitioners have not shown that Seshadri discloses or renders obvious a “bridge” that is “lockable to prevent,” and “unlockable to enable,” user access to “each of the sandboxes and the user-definable programmable electronic circuits therein.” EX1001, cl. 1.

74. I understand that Petitioners theory is that Seshadri's shell logic locks or unlocks to prevent or enable user access to sandboxes. However, in my opinion, Seshadri discloses no such functionality. Instead, Seshadri discloses only that the client-configurable circuit's access to other components of the peripheral device can be restricted. Seshadri does not disclose that access to the sandboxes is ever restricted, as required by each claim.

75. Petitioners begin by purporting to identify their alleged bridge: "Seshadri's shell logic includes port functionality that enables communication with the client-configurable circuits," and the "port" in turn "comprises functionality to perform 'bridging functions' between the PCIe interface and the client-configurable logic." Pet., 28; ID, 21-22 (finding that Petitioners' alleged "router with a port" and "bridge" was the "shell logic" generally). Petitioners assert that this "[t]he bridge is lockable/unlockable to prevent or enable user access to each sandbox." *Id.*, 29. I do not see support for this in the specification.

76. Petitioners first argue that "the shell logic includes functionality to 'isolate a configuration access port'" and "this functionality is locked/unlocked to disallow or allow configuration of client-configurable circuits." Pet., 29 (citing EX1004, 6:11-41, 12:10-31; EX1003 ¶ 88). But I do not see support in the cited portions of Seshadri for Petitioners' argument. While Seshadri states at 6:11-41 that the shell logic may "isolate a configuration access port," it explains in the very same

sentence that it is “so that the customer logic cannot cause the client configurable circuit to be reprogrammed without using services provided by the computing service provider (*e.g.*, the shell logic).” EX1004, 6:11–41. The customer logic cannot reprogram the client configurable circuits “without using [the shell logic].” *Id.* In other words, the customer logic can always reprogram the client configurable circuits—access to the client configurable circuits is never “prevented”—it just cannot circumvent the shell logic to do so.

77. A POSITA would understand that it makes sense for the client configurable circuits to only be accessed via the shell logic, and that Seshadri does not suggest locking the shell logic or preventing access to the sandboxes. The shell logic acts as a security layer controlled by the computing service provider, where it “can potentially increase the security, performance, privacy, and/or availability of the computing resources.” EX1004, 3:41-45. It “can provide a framework for the client configurable circuit to work within.” *Id.*, 3:50-51. Thus, the shell logic is a practical means to make access to and reprogramming of the client configurable circuits easier and safer for the rest of the system—it does not prevent access to those circuits.

78. Based on my analysis, the only other portion of Seshadri that Petitioners cite for this argument, 12:10-31, says nothing about isolating the configuration access port, let alone that access to the alleged sandboxes can be prevented or

allowed via locking and unlocking a bridge. Pet., 29 (citing EX1004, 12:10-31 (discussing that a client configurable circuit may be reconfigured by a client without identifying any restrictions)).

79. Petitioners next argue that “[w]hen configuration is requested by the user, the shell logic checks ‘customer account permissions’ to reject or accept the request to unlock the reprogramming functionality.” Pet., 30 (citing EX1004, 9:48-10:15). Again, I believe that Petitioners mischaracterize Seshadri to read limitations into it that do not exist.¹ Seshadri makes only a single reference to “customer account permissions,” and that sentence is fatally short on details: “First, the customer account permissions can be checked.” EX1004, 9:54. A POSITA would not

¹ I note that Petitioners state in a footnote that Patent Owners alleged that these limitations are satisfied when “the user must log in” to verify account permissions before writing an image. But in my opinion, the claims are not met simply by disclosing the idea of “log[ging] in.” The claims require that the bridge can be locked to prevent access to the sandboxes. Thus, logging only discloses this element if the logging in can lead to the bridge can be locked to prevent access to the sandboxes. Pet. 30, EX1021, 000034-35, 000055-56. Only showing that the system accepts login credentials is not sufficient to establish unpatentability; logging in by itself does not disclose or suggest the claimed locking and unlocking.

understand that single line to disclose a “bridge” that is “lockable to prevent,” and “unlockable to enable,” user access to “each of the sandboxes and the user-definable programmable electronic circuits therein” for multiple reasons.

80. First, there is no disclosure or suggestion of preventing access to the client configurable circuits as a result of checking the “customer account permissions.” Petitioners assert the customer account permissions are checked to “reject or accept the request to unlock the reprogramming functionality,” Pet., 30, but Seshadri makes clear that, contrary to Petitioners’ assertions, permissions are checked to confirm that the customer logic executed on the client configurable circuits does not negatively impact *the host* or impermissibly access *other* resources or modules. Seshadri explains that “customer account permissions can be checked” “*in order to protect the host* during a transition that replaces the first customer logic with the second customer logic.” *Id.*, 9:48-53 (emphasis added).

81. Seshadri further explains that the *host* is vulnerable while new customer logic is executed on the client configurable circuits, because the “the protection logic within the reconfigurable shell may be temporarily inactive.” *Id.*, 9:29-47. Crucially, the “protection logic” is “used to ensure that the customer logic within the client configurable circuit in a peripheral device would not be used maliciously *to impact the host device or to cause security breach.*” *Id.* (emphasis added). Thus, Seshadri discloses that account permissions are checked to determine whether the alleged

sandboxes should be prevented from accessing other aspects of the system—**not** to determine whether access to the alleged sandboxes should be prevented.

82. After asserting that the shell logic either rejects or accepts unlocking of the reprogramming functionality, Petitioners cite Seshadri as saying “. . . Subsequently . . . the customer logic can be configured on the client configurable circuits and activated using the reconfigurable shell,” Pet., 30 (citing EX1004, 9:48-10:15—implying that the customer logic can only be configured on the client configurable circuits if there is a successful check of customer account permissions. But the “subsequently” in this quote from Seshadri refers to the preceding sentence where “[e]xisting customer logic [the previous customer logic] within the client configurable circuit in the peripheral device can [] be cleared or deactivated.” EX1004, 9:65-10:15. All Seshadri says is that after the old customer logic has been cleared out, the new customer logic can “subsequently” be loaded. There is no disclosure of locking the shell logic to prevent access to the client configurable circuits

83. Second, there is no disclosure or suggestion that it is the shell logic (the alleged “bridge”) that checks the “customer account permissions”; Seshadri is silent on what performs the “check[.]” But the full context of Seshadri makes clear it would not be obvious to a POSITA for the shell logic to perform the check—quite the opposite. Seshadri discloses that when a client chooses to execute new customer

logic on the client configurable circuits, “in many cases, the shell logic function (or circuit) on the reconfigurable shell may also be replaced by a new shell logic function (or circuit) to work with the second customer logic.” EX1004, 9:37-40. The shell logic can be, and often is, replaced as a part of executing new customer logic to the client configurable circuits. Thus, a POSITA would not have found it obvious for the shell logic to check account permissions, when that shell logic is itself in the process of being replaced. Instead, a POSITA would understand that it would more naturally be the host that checks account permissions.

84. Petitioners next argue that “Seshadri locks/unlocks the bridge to control user access to programming and ‘increase the security, performance, privacy, and/or availability of the computing resources’ in the FPGA.” Pet., 30 (citing Ex-1004, 3:41-64). Once again, Petitioners quote some portions of Seshadri while tacking on the claim language without support. Notably, Petitioners’ quoted portion of Seshadri states in full that: “A computing service provider can potentially increase the security, performance, privacy, and/or availability of the computing resources by wrapping or encapsulating the client configurable circuit within a shell logic provided and controlled by the computing service provider.” Ex-1004, 3:41-45.

85. If anything, the cited portion of Seshadri undercuts Petitioners’ previous argument. Seshadri states that the shell logic, as part of its purported bridging functionality, can ensure that the client configurable circuit “cannot directly

control the signaling on the local interconnect” and will “prevent commands from being sent to undesired locations.” EX1004, 3:54-64. This further demonstrates that Seshadri is concerned with preventing the client configurable circuit’s unauthorized access to other portions of the system, not with preventing access to the client configurable circuits themselves. *See also id.*, 2:8-35, 3:41-45, 3:61-64, 9:24-28 (“The reconfigurable shell can also include protection logic used to ensure that the customer logic within the client configurable circuit in a peripheral device would not be used maliciously to impact the host device or to cause security breach.”).

86. Recognizing that Seshadri does not disclose this element, Petitioners make a last-ditch obviousness argument. Petitioners’ argument reads in full: “Moreover, this would have been obvious because Seshadri teaches that the shell logic includes routing functionality with a port for enabling communications to the client-configurable circuits (*supra* §V.A.3[1b]), and that is what the bridging functionality performs for reconfiguration communications.” Pet., 30-31.

87. As is apparent to a POSITA, Petitioners do not make any substantive argument—they provide no reason, analysis, or evidence that locking the bridge to prevent access to the sandboxes would be obvious in the face of Seshadri’s silence. And their implication that the claimed locking functionality would be obvious for any “bridging functionality [] for reconfiguration communications” is wrong. Seshadri discloses that any bridging functionality in Seshadri’s shell logic is to

enable easier access to the client configurable circuits and to prevent the client configurable circuits from accessing unauthorized portions of the system. In light of Seshadri's clear disclosure of these functionalities, and failure to disclose the claimed functionalities, those claimed functionalities would not have been obvious. Moreover, it would not be obvious to a POSITA because implementing such locking functionality would undermine the very purpose of Seshadri's bridging logic (*i.e.*, maintaining routing paths for configuration access), thereby requiring a redesigning of the shell architecture rather than a routine modification.

b. Seshadri Does Not Disclose or Render Obvious that the Alleged “Router with a Port” are “Hardware-Based.”

88. In my opinion, Petitioners do not show that Seshadri contains a “hardware-based” “router with a port,” as required by limitations 1[b] and 10[b]. EX1001, cl. 1, 10. Petitioners' argument for why all of Seshadri's shell logic (Petitioners' alleged “router with a port”) is “hardware-based” is simple (but fundamentally flawed): (1) the shell logic maps to the claimed “router with a port,” (2) the shell logic is implemented in an FPGA and therefore, (3) the shell logic is “hardware-based.” Pet., 26. But the fact that functionality is on an FPGA does not mean the functionality is “hardware-based” under the correct construction.

89. To begin, the '943 patent makes clear that a “hardware-based” router with a port means that it “lacks a processor.” *See supra* § VII.b. Petitioners do not even allege that Seshadri's alleged “router with a port” “lacks a processor.”

90. Even had Petitioners alleged the shell logic is executed without a processor, I do not believe that Seshadri offers support for such an allegation. FPGAs contain processors on which software can perform routing and port functionalities. Thus, while FPGAs do include hardware-based portions, they also have processors and software-based functionalities implemented with those processors. The fact that the shell logic is on an FPGA does not mean that it is hardware-based. Seshadri does not disclose or suggest that any or all of the shell logic (Petitioners' alleged "router with a port") is executed without a processor.

91. If anything, I believe that Seshadri indicates the opposite: the shell logic is executed on a processor (and is thus not "hardware-based"). Seshadri's shell logic is a "wrapper" layered over the configurable circuits. EX1004, 2:44-59. Seshadri states that the client configurable circuit "can be programmed with a shell logic that may be provided and controlled by a computer service provider," indicating that the shell logic may be software program implemented by the computer service provider to "increase the security, performance, privacy, and/or availability of the computing resources." *Id.*, 3:24-27, 3:41-45.

92. Crucially, it is not the shell logic itself that is programmed in configurable circuits, but rather the shell logic "wraps" around the circuits to perform software operations. Seshadri's figures further undercut Petitioners' assertion that Seshadri discloses that the shell logic is implemented without

processors. Figure 3 shows that the shell logic is not implemented in the “client configurable circuits”—and gives no indication that the shell logic is otherwise implemented without a processor:

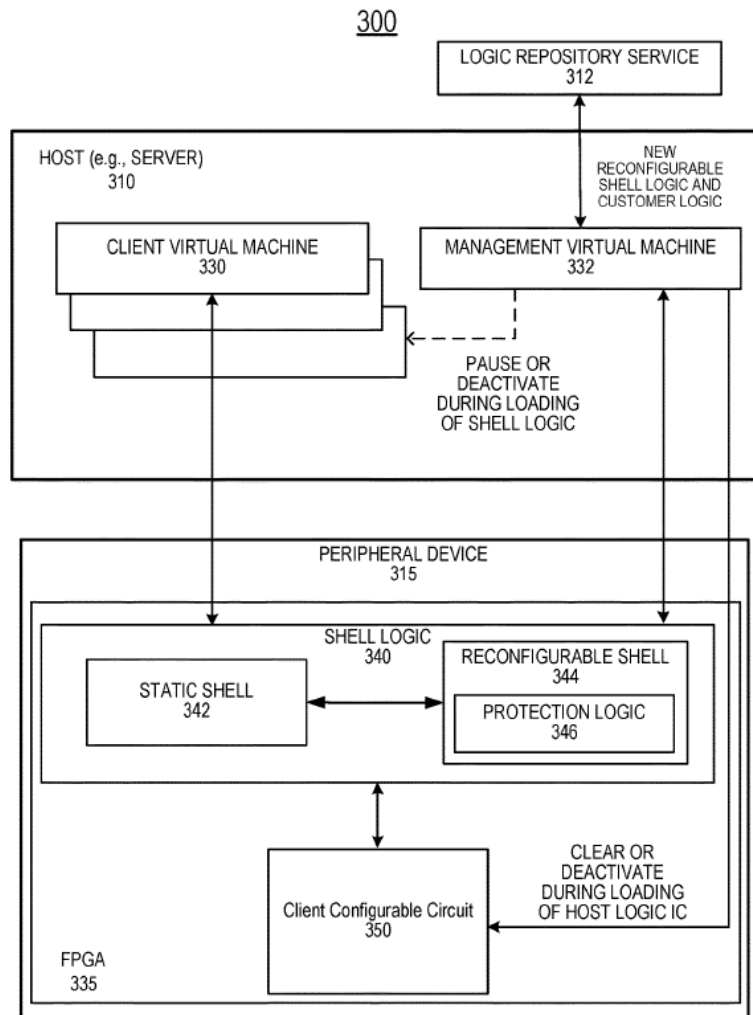


FIG. 3

EX1004, FIG. 3. Figure 4 similarly gives shows the shell logic separate from the client configurable circuits and again provides no indication that the shell logic is executed without a processor:

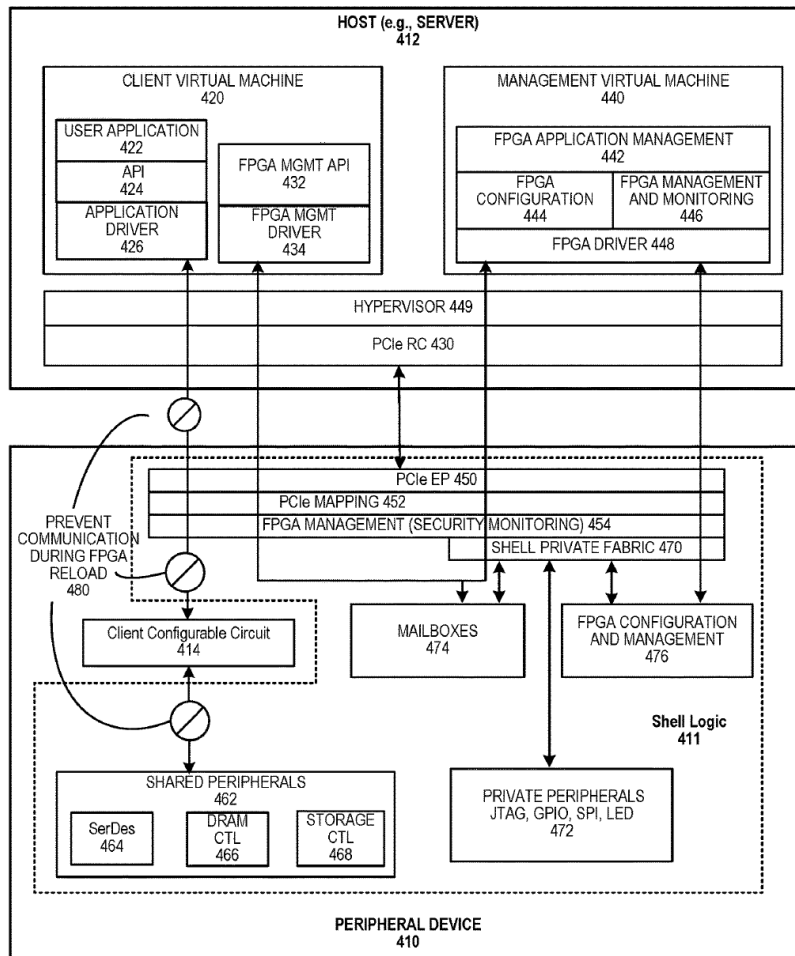


FIG. 4

Id., FIG. 4.

93. I understand that Petitioners also argue that the shell logic is hardware-based because it is implemented in the hardware programmable fabric of the FPGA, which they equate to the “client configurable circuits.” Pet., 27. As an initial matter, Petitioners do not explain why this assertion means that the shell fabric is executed without a processor and thus “hardware-based.”

94. In any event, Petitioners assertion is wrong; the shell logic is not implemented in the client configurable circuits. The client-configurable circuits are the portions of the device that may be configured **by the user** (*i.e.*, the sandboxes). As shown above, Seshadri and its figures make clear the shell logic is **not** implemented in the client configurable circuits. *See* EX1004, FIG. 3 and 4) (showing the shell logic separate from the client configurable circuits). At no point does Seshadri state that the shell logic is in the client-configurable circuits. Rather, the shell logic wraps around configurable regions of the device to enable certain functionality.

95. Petitioners' own evidence further demonstrates that Seshadri does not disclose that the shell logic is implemented in the client-configurable circuits. Petitioners cite Seshadri at 2:3 and 3:24 for support that the shell logic may act on the client-configurable circuit, but Seshadri is clear that the shell logic is separate from the client-configurable circuit and only wraps around it to provide the purported routing functionality. *Pet.*, 27. Seshadri explains that "[t]he client configurable circuit may be programmed with a shell logic, which may be provided and controlled by a computing service provider." EX1004, 2:2-5; 3:24-27 ("In certain implementations, the client configurable circuit can be programmed with a shell logic that may be provided and controlled by a computing service provider."). The client-configurable circuit may be programmed "with" the shell logic. A

POSITA would understand that the shell logic thus accompanies the client-configurable circuit; it is separate from the client-configurable circuit. This is made even clearer in the figures cited above where the shell logic sits outside and separate from the client-configurable circuits.

96. Further, Petitioners' argument ignores that, fundamentally, all devices necessarily use hardware, but that does not mean that all of their functionalities are implemented in hardware. A POSITA would understand that the distinction between software, firmware, and hardware-based "router with a port" in the context of the '943 patent is whether the claimed functionality is implemented through physical electronic components, like circuits, logic gates, and dedicated chips, that are designed for a specific purpose, or runs on programs, algorithms, and instructions executed by general-purpose processors. A functionality is software-based when it is implemented using a processor—and such software-based functionality may still exist on a PLD, FPGA, or ASIC.

97. Dr. Alpert makes the mistake of conflating these points. He provides a single sentence on this issue: "[b]ecause the shell logic is implemented using electronic circuits configured in the hardware of an IC, it is hardware-based." EX1003, ¶78. But as discussed above, using electronic circuits configured in the hardware of a chip does not mean that the device is hardware based if its functionality is otherwise provided by firmware or software. A POSITA would

recognize that all executable logic, whether software, firmware, or hardware, ultimately runs on electronic circuits, and the relevant distinction is not whether circuits are involved, but whether the functionality is implemented directly in fixed hardware or expressed as instructions stored in memory and executed by a processor. A “router with a port” that may be updated, replaced, or modified without redesigning the underlying chip is understood in the context of the ’943 patent to be software- or firmware-based, not hardware-based, much like the shell logic that operates through stored instructions interpreted by a processing element.

98. In my opinion, Petitioners make the same mistake when discussing the “port” within the router. Petitioners implicitly assume that the router is in hardware, thus the port is in hardware, but a POSITA would assume that Seshadri’s alleged “port” is implemented in software because modern networking devices rely heavily on virtual ports. In networking systems, ports are considered hardware when they are the physical connectors on a router and use physical cables, but they are in software when functioning as software constructs for directing traffic to downstream processes without physical connections. Given this well-understood distinction and the trend toward virtualization and software-based routing around the priority date, a POSITA would have expected Seshadri’s port to be implemented in software unless the reference expressly stated otherwise. If the port were in hardware, a POSITA would expect Seshadri to disclose physical connections between the router

and downstream sandboxes. The reference is silent on this, and Petitioners only point to exemplary mailbox functionality to show that there is some general port functionality in the system, without tying it back to the claim language. Pet., 26. Even where the router is within the programmable fabric of an FPGA, the port which enables communication between regions may be in software.

99. Based on my review of the Petition and Dr. Alpert's declaration, Petitioners have not shown that Seshadri discloses a "hardware-based" router with a port. At most, they show that Seshadri has routing functionality in the shell logic that is executed on an FPGA, but not that the router functionality is implemented without a processor.

2. Claims 5 and 14

a. Seshadri Does Not Disclose or Render Obvious that the Claimed "Bridge" with a "Security Layer."

100. In my opinion, Petitioners have not shown that Seshadri discloses a bridge having a security layer to authenticate a user or verify a permission to reconfigure one or more sandboxes, as required by claims 5 and 14. Petitioners contend that Seshadri's shell logic carries out "security functionality" that verifies whether a user's customer account has permission to reconfigure the client-configurable circuits. Pet., 45 (citing EX1004, 9:48-10:15). But there is no evidence tying this customer permission verification to the reconfiguration of the client-configurable circuits.

101. As discussed above, the only evidence that Petitioners provide for these claims is that the shell logic purportedly verifies customer account permissions, not that those permissions are verified for purposes of reconfiguring the sandboxes. The '943 patent describes that the bridge has a security layer designed to protect system circuitry, including by authenticating users before they are permitted to reconfigure the sandboxes. EX1001, 5:43-45. Seshadri only establishes that customer account permissions can be checked before loading customer logic as a means to protect the host or impermissible access to other resources. EX1004, 9:48-10:15. Put simply, there is no disclosure or suggestion of preventing access to the client configurable circuits as a result of checking the "customer account permissions."

B. Petitioners Fail to Prove their Combination of Atsatt-113 and Biederman (Grounds 3 and 4) Disclosed or Renders Obvious Claims 1–5, 9–14, and 18.

102. In my opinion, even if Atsatt-113 and Biederman were combined, the combination does not disclose or render obvious several claim limitations. Atsatt-113 does not disclose or render obvious a bridge that is lockable or unlockable, nor are its global resources a router with a port.

1. Claims 1 and 10

- a. Atsatt-113 Does Not Disclose or Render Obvious that the Alleged “Bridge” is “Lockable to prevent user access,” and “Unlockable to enable user access” “to each of the sandboxes and the user-definable programmable electronic circuits therein.”**

103. In my opinion, Petitioners have not shown that Atsatt-113 disclosed or renders obvious a bridge that is lockable and unlockable to prevent or allow user access to sandboxes, as required by limitations 1[d] and 10[e]. EX1001, cl. 1, 10. I observe that Petitioners map this locking/unlocking to Atsatt-113’s “masking” functionality, which provides each alleged sandbox (Atsatt-113’s partitions²) with a “mask” that dictates what resources the sandbox may and may not access. When a user seeks to reconfigure a sandbox, the mask will block or allow the reconfigured partition for accessing prohibited resources. *Id.* Petitioners argue that Atsatt-113’s “interconnection resources” are the claimed “bridge” and enforce the mask and what resources each sandbox may access. Pet., 65-66.

104. Petitioners’ mapping suffers from two flaws. First, Atsatt-113’s “masks” do not prevent or allow access to the sandboxes. A POSITA would understand that masks are fundamentally different. Users can always access the

² Atsatt-113 explains that partitions are reconfigurable regions of the programmable fabric that group together resources on the programmable device. EX1006, ¶¶ 5, 39.

partitions (*i.e.*, the sandboxes). Masks determine what resources the sandboxes can access, but not whether the sandboxes themselves can be accessed. Atsatt-113 does not disclose that access to sandboxes themselves can be prevented by masks. Second, Atsatt-113 does not disclose that the alleged bridge (“interconnection resources”) can be locked or unlock or that they are what enforces the “masks.”

105. Thus, in my opinion, Petitioners have failed to show that Atsatt-113 discloses or suggests a bridge that is lockable and unlockable to prevent or allow user access to sandboxes.

(1) Atsatt-113’s Masks Do Not “Prevent User Access” Or “Enable User Access” “to each of the sandboxes.”

106. In my opinion, Atsatt-113 teaches a fundamentally different form of access control over sandboxes than what is claimed in the ’943 patent. Claims 1 and 10 of the ’943 patent specify require a bridge that can be locked to prevent access to the sandboxes EX1001. 5:19-38. Just as with a physical bridge, the ’943 patent’s bridge can be shut down so that whatever lies beyond it (in this case the sandboxes) cannot be reached.³

³ Claim 4 also requires that the bridge can be locked to prevent access to the bridge itself—effectively shutting down the bridge at its entrance.

107. Atsatt-113, on the other hand, describes a quite different system. In Atsatt-113, each partition is given a mask by a configuration controller “that indicates resources to which the partition is associated.” EX1006, ¶¶ 46-49, 56. Instead of a bridge, Atsatt-113’s “masks” are effectively forms of identification (such as car registrations) that state what resources the partition may access (for example whether they may drive on highways). The ’943 patent and Atsatt-113 thus present fundamentally different methods of controlling access: The ’943 patent requires a bridge through which all traffic must flow, which can be shut off to prevent all or a portion of the traffic; while Atsatt-113 describes a system in which traffic can travel without restriction and unauthorized access is prevented by each partition identifying (via “masks”) what resources they are allowed.

108. Most notably, Atsatt-113 does not disclose any way to prevent access to “each of the sandboxes.” Petitioners map Atsatt-113’s “partitions” to the claimed sandboxes. Pet., 62-63 (“Atsatt-113 teaches that the partitions are sandboxes”) (“each sandbox (e.g., the user partitions)”). But Atsatt-113 is clear that masks dictate what resources the partition may access—not which partitions may themselves be accessed, or by whom. EX1006, ¶ 5 (“A configuration controller, before or during runtime, may generate a mask for a partition that indicates resources to which the partition is associated.”), ¶ 26 (“As discussed in further detail below, embodiments of the present disclosure relate generally to partial reconfiguration of only a portion

of a programmable fabric using a mask to determine which resources may be changed by the partition.”), ¶¶ 41-57. “[A]llowed’ masks” indicate “all resources that are owned by a partition” while “an inverted allowed masks (*e.g.*, ‘Not Allowed’)” shows those resources that are not. But fundamentally, masks do not stop users from accessing the *sandboxes* (partitions).

109. I do not believe that Petitioners’ citations to Atsatt-113 show that masks can prevent access to partitions. Petitioners repeatedly allege that “the mask mechanism is used to enable or prevent user access to each partition” but that is made up out of whole cloth. For example, Atsatt-113, at ¶ 5, states that “[a] configuration controller, before or during runtime, may generate a mask for a partition that indicates resources to which the partition is associated,” but this only shows that partitions may be limited to certain resources, not that users are prevented from access those other resources. EX1006, ¶ 5.

110. Petitioners’ own arguments exemplary the mismatch between Atsatt-113 and the ’943 patent. From a starting point, Petitioners are wrong that Atsatt-113 discloses using masks to prevent access to sandboxes. But Petitioners’ argument makes no sense even on its own terms. Petitioners argue that access to “each of the sandboxes” can be prevented via masks. Pet., 64-67. But even with Petitioners’ (incorrect) understanding of Atsatt-113, that would require the configuration controller modifying the ownership attributes associated with each mask. Atsatt-113

does not disclose or suggest any such bizarre scenario. Nor would such a scenario make any sense to a POSITA. I believe that modifying all partition masks to prevent access to any resources ties access control to distributed, user-specific identifiers, making system-wide locking/unlocking nearly impossible. Atsatt-113's masks are used to ensure that partitions do not change resources that other partitions are relying on—the are ill-suited for preventing access to all resources.

111. By contrast, when access to the sandboxes is governed by the ability to lock or unlock the bridge itself (as in the '943 patent), the system operator gains centralized and immediate authority over user access without needing to modify individual user credentials. In such an architecture, the operator can allow or deny access at any time without altering the underlying user-mask ownership structure. Ultimately, Atsatt-113 does not disclose or suggest anything close to the claimed arrangement.

**(2) Atsatt-113's "Interconnection Resources"
Cannot Be Locked Or Unlocked And Do Not
Enforce The "Masks."**

112. In my opinion, Petitioners' argument fails for a second, independent reason: the claims require that it is the bridge that can be locked to prevent access to the sandboxes, but Atsatt-113 does not disclose that the alleged bridge ("interconnection resources") can be locked or unlock or that they enforce the "masks." The "interconnection resources" in Atsatt-113 are "used to route signals,

such as clock or data signals, through the FPGA 40,” EX1006, ¶ 30, and “may act as a network between the device controller 60 and each sector controller 58,” *id.*, ¶ 33. In short, the “interconnection resources” can route signals between various portions of Atsatt-113’s system.

113. Based on my analysis, however, there is no suggestion that the “interconnection resources” can be locked or that they enforce the resources that sandboxes can access. In fact, Atsatt-113 explains that it is the “central authority that assigned the partitions” or the “device configuration system” that enforces masks—not the “interconnection resources.” *Id.*, ¶¶ 26, 42, 56, 53 (“In some embodiments, when the configuration system detects a violation, the configuration may halt configuration of the programmable device.”).

114. None of Petitioners’ evidence that I reviewed demonstrates that Atsatt-113’s purported bridge in the interconnection resources may be being locked or unlocked or that they enforce the masks in any way. Petitioners assert that “[w]hen [communications] pass over the interconnection resources, they are modified based on the applicable mask state.” Pet., 66 (citing EX1006 ¶ 41). That is not what Atsatt-113 discloses. Paragraph 41 of Atsatt-113 does not describe the interconnection resources or any bridging functionality. Instead, it only describes the masks and describes that they “define the exact resources that are to be left un-modified.”

EX1006 ¶ 41. There is no explanation that the interconnection resources are responsible for “enforcing” the mask by locking or unlocking itself.

115. Even worse, Petitioners contend Atsatt-113 teaches sending the mask to the interconnection resources for the express purpose of enforcing the mask when distributing configuration data to partitions. Pet., 66-67. But this is missing from the disclosure. Atsatt-113 does not teach that masks are sent to the interconnection resources to “enforce” partition restrictions, nor does it describe mask authentication as determining whether a user has access to particular partitions. Instead, Atsatt-113 makes clear that mask enforcement occurs *entirely within the partition maintenance circuitry*. EX1006, ¶ 66.

116. The interconnection resources, at most, act as a routing fabric, not an enforcement mechanism. For example, ¶ 81 discusses that “the validated or verified configuration data 224 is sent to the interconnection resources 46 to be distributed to one or more sectors 48 of the FPGA 40,” but ¶ 66 describes that protection against unauthorized reconfigurations is enforced by the partition maintenance circuitry. EX1006, ¶¶ 66, 81. Atsatt-113 even contemplates embodiments where the mask is not sent to the interconnection resources at all, so there is no support for Petitioners’ argument that these interconnection resources are purportedly responsible for mask authentication.

117. I observe that Petitioners further suggest that the interconnection resources become locked or unlocked when a user's key is authenticated. Pet. 66. But again, none of their cited paragraphs shows that the interconnection resources can be locked or unlocked, or even that they are responsible for "enforcing" the mask in any manner. Paragraph 26 merely states that masks may be assigned by a central authority (*e.g.*, a server owner). Paragraph 59 only describes users sending data streams to reconfigure partitions. And paragraph 81 explains that authenticated data will be sent to the interconnection resources for distribution to one or more FPGA sectors. EX1006 ¶¶ 26, 59, 81. But none of these passages address any mechanism by which the interconnection resources or associated bridge functionality are locked or unlocked. In fact, Atsatt-113 suggests the opposite as the "partition maintenance circuitry may detect errors in the data before being sent to the interconnection resources," indicating that the interconnection resources have no form of access control. *Id.*, ¶¶ 79-81.

118. In summary, Petitioners only show that Atsatt-113 discloses mechanisms for preventing or permitting reconfiguration of individual sandboxes. This is substantially different than locking or unlocking the bridge to prevent user access to individual sandboxes, which is what the '943 patent requires.

b. Atsatt-113 Does Not Disclose or Render Obvious the Claimed “Router with a Port” In The Alleged “First Region.”

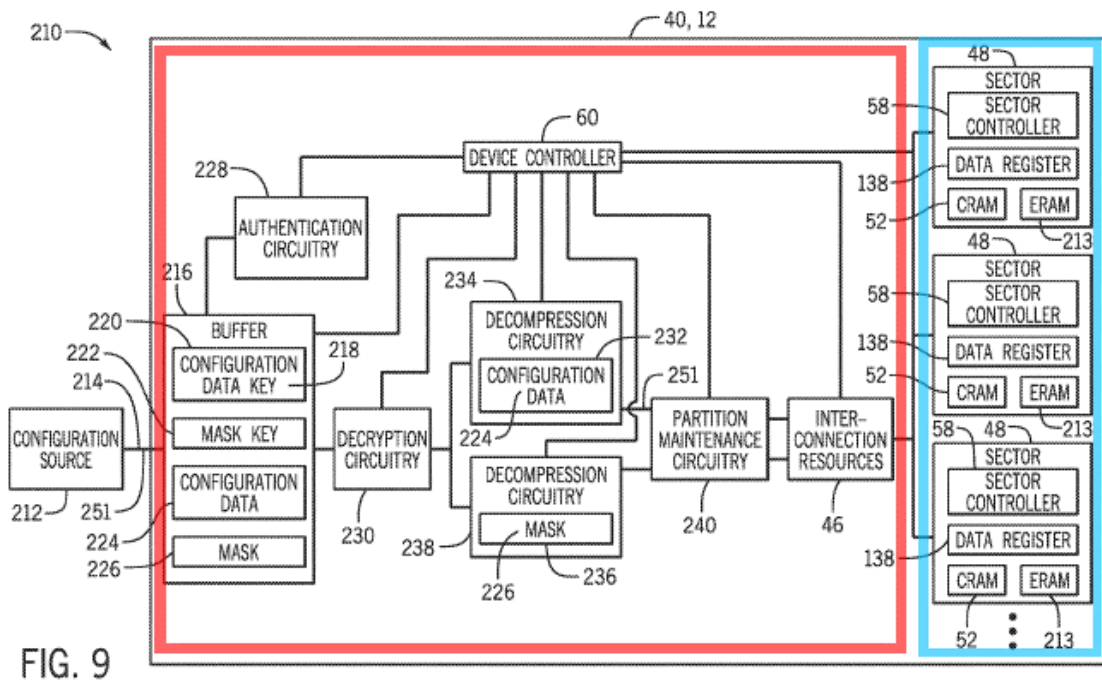
119. In my opinion, Petitioners have not shown that Atsatt-113 disclosed or renders obvious the claimed “router with a port,” as required by claim limitations 1[b] and 10[b], for multiple reasons. EX1001, cl. 1, 10. Petitioners unambiguously map the claimed “router with a port” to Atsatt-113’s “global resources.” But the global resources are only resources shared between the partitions in the second region, not the first region. Pet., 59, 61; EX1006, ¶ 55. To a POSITA, Petitioners’ have thus failed to show Atsatt-113 discloses a “router with a port” in the first region, as required by the claim.

120. I disagree with Petitioners that Atsatt-113’s global resources are in the first region. Atsatt-113 explains that global resources are resources, such as input/output ports and serial interfaces, which are shared *between the partitions in the sectors* comprising the second region of the system:

Global resources are often configured once and not updated by partition updates. For example, global resources may include input/output ports (e.g., 1048), interfaces (e.g., high-speed serial interface—HSSI), hard processor systems (HPS), a clock network, and/or other resources *shared between the partitions*. Often, they are defined by a base partition at the time the set of partitions is defined. For global resources that are dynamically reconfigurable by the core logic, the base partition may claim the resources that provide access to the reconfiguration pathways.

EX1006, ¶ 55 (emphasis added). Atsatt-113 describes that, “[i]n some embodiments, the partitions may include a base partition that is assigned the global resources.” *Id.*, ¶ 56. It goes on to state that the global resources “include resources that are shared between the sectors 102A-110H and outside of the physical regions of the sectors 102A-110H and/or partitions of the programmable fabric 100.” *Id.*, ¶ 39. Thus, the resources are global to the sectors, not the entire device, meaning they cannot comprise the router and port as they are not in the first region and are not responsible for transmitting data from the configuration source to the sandboxes.

121. Dr. Alpert’s opinion appears to confirm this understanding. Based on my review of his deposition, he was explicit that he mapped the global resources to the claimed router. EX2011, 35:22-36:6, 38:13-16. 45:19-46:4, 57:7-13. Further, his annotated versions of Figures 3 and 9 undermine Petitioners’ argument. Figure 9 shows that the configurable partitions are in the second region (which Dr. Alpert has shown in blue). Figure 3 of Atsatt-113—which shows only the configurable partitions of the second region—shows that the “global resources” (highlighted in yellow) sit among the partitions in the second region, not within the first region:



Id., ¶ 179 (citing EX1006, FIG. 9).



EX1003, ¶ 182 (citing EX1006, FIG. 3).

122. As a POSITA, I do not believe that this mapping is consistent with the architecture of the '943 patent. The '943 patent provides that the system contains a bridge which “is a module that sits on a router port, for purposes of crossing the I/O type or protocol.” EX1001, 4:57-59. The inventors designed a system where the router, bridge, and port sit *separate* from the second region containing the sandboxes so that this first region cannot be reconfigured by users when they reconfigure the sandboxes. Petitioners, however, present an analysis in which these regions are indiscriminately grouped together within the second region. Pet., 58-64.

123. Petitioners cite to some functionalities performed by the “device controller 60” and “interconnection resources” to show that the “global resources” maps to the claimed “router with a port.” But Atsatt-113 is clear that the device controller and interconnection resources are *not* part of the global resources. Pet., 58. For example, Atsatt-113 describes the “interconnection resources” separately from the global resources and states that “[t]he programmable fabric 100 also includes global resources 112, 114, 116, and 118.” EX1006, ¶ 39. Atsatt-113 also does not describe the “device controller” as part of the global resources. *Id.*, ¶ 55.

124. This highlights Petitioners’ fundamental issue: their alleged router sits in the second region, not the first. Similarly, Petitioners separately identified functionality within the interconnection resources as the “bridge” in limitations 1[d] and 10[e], discussed above. The bridge must be a part of the router, yet Petitioners’

alleged bridge (interconnection resources) is not within their alleged “router with a port” (global resources).

c. Atsatt-113 Does Not Disclose or Render Obvious that the Alleged “Router With a Port” are “Hardware-Based.”

125. In my opinion, Petitioners have not shown that Atsatt-113 discloses or renders obvious the claimed “router with a port” that are “hardware-based,” as required by claim limitations 1[b] and 10[b]. EX1001, cl. 1, 10. Petitioners unambiguously map the claimed “router with a port” to Atsatt-113’s “global resources,” which they contend includes the device controller and interconnection resources. Pet., 59, 61. But their only support for the requirement that these devices are hardware-based is that they are “are implemented on the hardware-programmable fabric of the FPGA.” Pet., 62 (citing EX1006, ¶ 39).

126. As discussed above, the ’943 patent makes clear that a “hardware-based” router with a port means that it “lacks a processor.” *See supra* § VII.b. Petitioners do not even allege that Atsatt-113’s alleged “router with a port” “lacks a processor,” nor do they reconcile the fact that FPGAs regularly contain processors on which software can perform routing and port functionalities. Atsatt-113 does not disclose or suggest that all of the global resources (Petitioners’ alleged “router with a port”) is executed without a processor. In fact, it expressly includes “hard processor

systems (HPS)” as an example of a global resource. EX1006, ¶ 55. A POSITA would understand that HPS is a system architecture that includes a dedicated processor.

127. Further, even though Petitioners also point to other components in the system, such as the device controller and interconnection resources, nothing in Atsatt-113 states that they are not implemented using processors executing software routines. In fact, Atsatt-113 expressly states that the device controller “may be implemented as state machines and/or processors,” and that the control program may reside in RAM so that “new routines” may be written to implement “new operations and functionality” within the FPGA fabric. EX1006, ¶ 32. It also describes that the device controller may execute computer-readable instructions, indicating it is a processor that does not meet the claim construction for a “hardware-based” router with a port. *Id.*, ¶ 58. This describes software-based behavior using a processor instead of fixed hardware logic. Likewise, the interconnection resources are described as a packet-based communication network between controllers, which is commonly orchestrated by embedded processors or engines.

2. Claims 4 and 13

128. In my opinion, Petitioners also fail to show that Atsatt-113 discloses claims 4 and 13. As discussed above, these claims teach that the bridge is lockable to prevent reconfiguration of the bridge and router while enabling reconfiguration of the one or more sandboxes. EX1001, cl. 4, 13. Petitioners contend that Atsatt-113

teaches this because the global resources, namely a “base partition,” are ““configured once and not updated by partition updates’ such as reconfiguration of the user partitions.” Pet., 75. However, this does not demonstrate that the bridge (the interconnection resources) is lockable. As a POSITA, I would not view configuring the system at setup to mean it is “lockable.”

129. As an initial matter, Petitioners make the same mistake discussed above: *i.e.*, they map the bridge and router to global resources. Petitioners point to the “base partition” within global resources for this limitation, claiming that the system’s routing and bridge functionality are “assigned to a base partition” within the system. Pet., 75. Patent Owners explained above that global resources are not a bridge, router, and port within a first region like the ’943 patent claims, but are instead a limited set of resources accessible to the sectors and partitions in the second region. EX1006, ¶¶ 30, 55. Seshadri provides examples such as “input/output ports,” “interfaces,” and “a clock network” because these global resources sit separately, between partitions, away from the first region. Because global resources are not in the first region, the base partition within those global resources cannot be the lockable bridge and router.

130. Setting aside that Petitioners fail to identify any bridge or router, I believe they are still wrong that the purported “base partition” satisfies the claim language. Petitioners contend that Atsatt-113’s masking functionality may establish

a “not allowed mask[]” for certain partitions, including the base partition in which the bridge and router allegedly reside. Pet. 75–76. But as I explained above, Atsatt-113’s masking functionality is not equivalent to locking or unlocking a bridge to prevent access to the sandboxes and cannot be mapped to the claimed bridge and router. *See supra* § VII.B.1.a.

131. Petitioners further argue that the base partition is “lockable” because it “is locked after initial setup.” Pet. 76. But a device being defined or configured at setup does not mean that it is lockable, it simply means that initialization has occurred. A lockable device, by definition, is one that is both capable of being locked and unlocked. Merely establishing a device’s configuration at setup does not demonstrate that its state can later be changed. Further, it does not disclose that the bridge is “unlockable” to allow access to the bridge, as the configuration at setup is inherently not later changed (unlocked).

132. Nor, in my opinion, does Atsatt-113’s disclosure support Petitioners’ position. Atsatt-113 explains that “[g]lobal resources are often configured once and not updated by partition updates” and that “[o]ften, [global resources] are defined by a base partition at the time the set of partitions is defined.” EX1006, ¶ 55. But the fact that global resources are “often” configured only once does not mean they are lockable to affirmatively prevent reconfiguration. Rather, Atsatt-113 describes a design choice or administrative practice by the system owner, not the presence of a

locking mechanism that prohibits reconfiguration of the bridge and router. At most, Atsatt-113 explains that masks may be generated to “describe the ownership” of certain partitions, but there is no discussion that masks may define ownership of the base partition such that it may be locked or unlocked.

3. Claims 5 and 14

a. Atsatt-113 Does Not Disclose or Render Obvious that the Claimed “Bridge” with a “Security Layer.”

133. In my opinion, Petitioners have not shown that Atsatt-113 discloses a bridge having a security layer to authenticate a user or verify a permission to reconfigure one or more sandboxes, as required by claims 5 and 14. Petitioners contend that Atsatt-113 teaches these claims because the “port and bridge functionality of the global resources” also “authenticates configuration data and masks to verify permission to reconfigure the partitions.” Pet., 76 (citing EX1006, ¶ 62). Petitioners specifically argue that “authentication circuitry” may authenticate the mask using, for example, “a secure has algorithm.” *Id.* However, Atsatt-113 only teaches authentication of a mask to ensure the mask is not corrupt, not that a user is permitted to reconfigure certain partitions.

134. As I discussed above, the ’943 patent describes that the bridge has a security layer designed to protect system circuitry, including by authenticating users before they are permitted to reconfigure the sandboxes. EX1001, 5:43-45. But in Atsatt-113, the only evidence that Petitioners provide shows how the mask is

authenticated using an authentication key or a secure hash algorithm, which are security tools to ensure that the mask and configuration data being received are legitimate. Figure 16, cited by Petitioners, shows that the configuration data, mask, and authentication keys are all provided at the same time by the user seeking to reconfigure a region. EX1006, FIG. 16. Thus, the purported bridge does not contain a security layer that compares the mask against stored account permissions, rather it accepts all data received by the user and may apply a hash algorithm, like SHA-1, which authenticates the data as the same data that was sent by the user.

C. Ground 5 Does Not Address The Problems In Grounds 1–4.

135. In ground 5, Petitioners rely on Tarafdar only for the limitations recited in claims 6 and 15. Petitioners do not rely on Tarafdar to teach or suggest any of the limitations challenged above. In my opinion, Tarafdar thus does not address the deficiencies in Grounds 1–4.

D. Ground 6 Does Not Address The Problems In Grounds 1–4.

136. In ground 6, Petitioners rely on Kulkarni only for the limitations recited in claims 7 and 16. Petitioners do not rely on Kulkarni to teach or suggest any of the limitations challenged above. In my opinion, Kulkarni thus does not address the deficiencies in Grounds 1–4.

E. Ground 7 Does Not Address The Problems In Grounds 1–4.

137. In ground 7, Petitioners rely on Atsatt-923 only for the limitations recited in claims 8 and 17. Petitioners do not rely on Atsatt-923 to teach or suggest

any of the limitations challenged above. In my opinion, Atsatt-923 thus does not address the deficiencies in Grounds 1–4.

XII. SIGNATURE

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed on this 29th day of January, 2026.

A handwritten signature in black ink, appearing to read "Paul S. Min", is written above a horizontal line.

Dr. Paul S. Min