

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

PATENT OWNERS' RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND	3
A.	Overview of the '943 Patent.....	3
B.	Overview of Petitioners' Prior Art.....	7
1.	U.S. Patent 10,776,142 ("Seshadri") (EX1004)	7
2.	U.S. Patent Application Publication No. 2018/0191642A1 ("Biederman") (EX1005).....	9
3.	U.S. Patent Application Publication No. 2019/0095113A1 ("Atsatt-113") (EX1006)	9
4.	"Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center" ("Tarafdar") (EX1007).....	12
5.	U.S. Patent No. 8,015,386 ("Kulkarni") (EX1008).....	12
6.	U.S. Patent Application Publication No. 2019/0041923A1 ("Atsatt-923") (EX1009)	13
C.	Alleged Unpatentability Grounds.....	13
III.	PROSECUTION HISTORY	13
IV.	PRIORITY DATE OF THE '943 PATENT	14
V.	LEVEL OF ORDINARY SKILL	14
VI.	RELATED MATTERS	15
VII.	CLAIM CONSTRUCTION	15
A.	"programmable logic device".....	15
B.	"hardware-based" "router with a port"	16

VIII. PETITIONERS HAVE NOT PROVEN THAT ANY CHALLENGED CLAIM IS UNPATENTABLE	17
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.....	17
1. Claims 1 and 10.....	17
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.”	18
b. Seshadri Does Not Disclose or Render Obvious that the Alleged “Router With a Port” are “Hardware-Based.”	27
2. Claims 5 and 14.....	35
a. Seshadri Does Not Disclose or Render Obvious that the Claimed “Bridge” with a “Security Layer.”	35
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.....	36
1. Claims 1 and 10.....	37
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.”	37
i. Atsatt-113’s Masks Do Not “Prevent User Access” Or “Enable User Access” “to each of the sandboxes.”	38

ii.	Atsatt-113’s “Interconnection Resources” Cannot Be Locked or Unlocked and Do Not Enforce the “Masks.”	41
b.	Atsatt-113 Does Not Disclose or Render Obvious the Claimed “Router with a Port” In the Alleged “First Region.”	44
c.	Atsatt-113 Does Not Disclose or Render Obvious that the Alleged “Router With a Port” are “Hardware-Based.”	49
2.	Claims 4 and 13.....	50
3.	Claims 5 and 14.....	53
a.	Atsatt-113 Does Not Disclose or Render Obvious that the Claimed “Bridge” with a “Security Layer.”	53
C.	Ground 5 Does Not Address The Problems In Grounds 1–4.....	54
D.	Ground 6 Does Not Address The Problems In Grounds 1–4.....	54
E.	Ground 7 Does Not Address The Problems In Grounds 1–4.....	55
IX.	CONCLUSION.....	55

PATENT OWNERS' EXHIBIT LIST

Exhibit	Description
2001	Text Order Denying AMD/Pensando's Motion to Stay Case (Judge Robert Pitman) in <i>Concurrent Ventures, LLC and XstreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.</i> , No. 1-24-cv-00335 (W.D. Tex. 2024)
2002	First Amended District Court Scheduling Order (Dkt. 77) in <i>Concurrent Ventures, LLC and XstreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.</i> , No. 1-24-cv- 00335 (W.D. Tex. 2024)
2003	Webpage printouts of Docket Navigator Time to Trial regarding the District Court for the Western District of Texas and Judge Alan D. Albright
2004	Webpage printout of "AMD Reports Fourth Quarter and Full Year 2024 Financial Results" dated March 4, 2025 (https://ir.amd.com/news-events/press-releases/detail/1236/amd-reports-fourth-quarter-and-full-year-2024-financial-results)
2005	AMD/Pensando's Invalidity Contentions (Cover Pleading) in <i>Concurrent Ventures, LLC and XstreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.</i> , No. 1-24-cv-00335 (W.D. Tex. 2024)
2006	AMD/Pensando's Motion for Reconsideration of Venue Order in <i>Concurrent Ventures, LLC and XstreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.</i> , No. 1-24-cv00335 (W.D. Tex. 2024)
2007	Order denying AMD/Pensando's Motion for Reconsideration of Venue Order in <i>Concurrent Ventures, LLC and XstreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.</i> , No. 1-24-cv-00335 (W.D. Tex. 2024)
2008	Declaration of Zachary L. Jacobs in Support of Motion for Admission Pro Hac Vice

NEW EXHIBITS

New Exhibit	Description
2009	Declaration of Dr. Paul S. Min in Support of Patent Owners' Response
2010	Curriculum Vitae of Dr. Paul S. Min
2011	Deposition of Dr. Alpert in IPR2025-00223, conducted January 16, 2026

Concurrent Ventures, LLC and XstreamEdge, Inc. (“Patent Owners”) respectfully submit this Patent Owners’ Response (“POR”) to 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”), and to the Board’s November 6, 2025 Institution Decision (“ID”) (Paper 14).

I. INTRODUCTION

The Petition fails to render obvious the inventions of the ’943 patent. Instead, Petitioners’ references disclose the same prior art systems disclosed and improved upon by the ’943 patent. Given these deficiencies, a POSITA would not have found the ’943 patent obvious at the time of the invention.

Claims 1 and 10 disclose a programmable logic device with a “first region” and “second region.” This first region includes a hardware-based or firmware-based “router with a port” that enables communication with the second region. The second region, in turn, includes sandboxes that contain user-definable programmable electronic circuits. Crucially, the port in the first region also comprises a bridge that is lockable to prevent user access to the sandboxes and programmable circuits, and unlockable to allow such access. In this way, there is a central communication bridge that can mediate user access to each of the sandboxes and the user-definable programmable electronic circuits. Claims 4 and 13 add that the bridge may lock access to itself while still allowing access to the sandboxes. Claims 5 and 14 further

add that the bridge has a security layer to authenticate a user or verify permission to reconfigure one or more of the one or more sandboxes.

Petitioners fail to demonstrate that the limitations of any of these claims is disclosed and/or obvious under any of their grounds. First, Seshadri and Atsatt-113 teach fundamentally different forms of access control to client-configurable circuits than the '943 patent requires. Seshadri teaches a “shell logic” (the alleged “router with a port” and “bridge”) that wraps around configurable circuits (the alleged “sandboxes”) to make access to those circuits easier, and to prevent those circuits from improperly accessing *other modules* within the device. Seshadri does not even contemplate the idea of mediating access to the alleged sandboxes, much less disclose the '943 patent specific approach to doing so (with a central bridge that can lock). Atsatt-113 similarly teaches a masking functionality (the alleged “locking” functionality) that dictates what resources the partitions (the alleged “sandboxes”) can access. Again, Atsatt-113 does not suggest that access to the alleged sandboxes is prevented, or that it is the alleged “bridge” (“interconnection resources”) that locks to prevent access.

Moreover, neither Seshadri nor Atsatt-113 disclose a *hardware-based* “router with a port.” The '943 patent is clear that a “hardware-based” router is one that lacks

a processor.¹ Yet Seshadri and Atsatt-113 do not disclose or suggest that their alleged “router[s] with a port” are implemented in hardware, without a processor. In fact, both suggest the opposite: their alleged routers are not implanted with hardware circuits; a POSITA would understand they are likely on processors.

Petitioners have thus failed to prove by a preponderance of the evidence that all claims of the ’943 patent are unpatentable.

II. BACKGROUND

A. Overview of the ’943 Patent

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 rapidly changing data formatting requirements or new data processing applications. EX1001, 1:8–21. For example, video processing changes rapidly, and while it is faster to process data in hardware,

¹ The Board invited Patent Owner to propose a construction of “hardware-based” if appropriate. ID, Paper 14 at 18. The ’943 patent specification provides a definitive definition of “hardware-based,” which Patent Owners propose now. The Board has not had an opportunity to weigh in on whether the references disclose a “hardware-based” “router with a port” under this new, correct construction. As explained in detail, they do not.

many new standards are constantly and rapidly introduced, resulting in the need for flexible solutions for data processing. *Id.*

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. The programmable device comprises a programmable logic device (“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. 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. *Id.* 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.*

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

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.

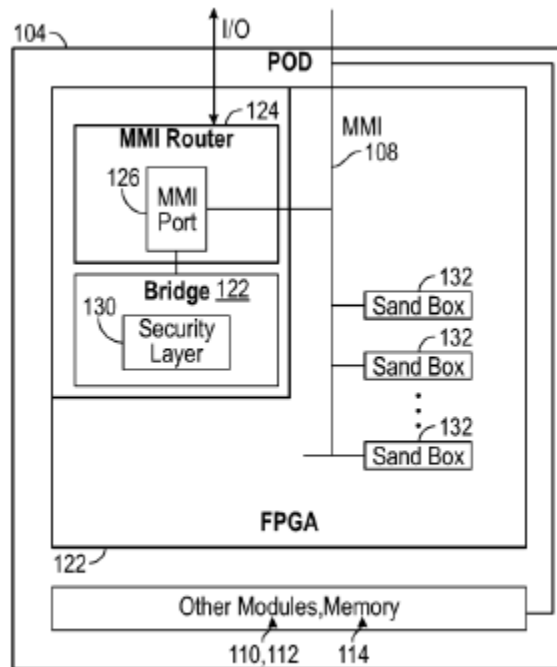


FIG. 3

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.*

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.*

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 combination 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. Overview of Petitioners' Prior Art

1. U.S. Patent 10,776,142 ("Seshadri") (EX1004)

Seshadri discloses techniques for reconfiguring programmable hardware in a cloud-computing environment while a client virtual machine 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. These devices, including FPGAs, contain both processors that can execute software and configurable circuits on which hardware logic can be executed. EX2009, ¶ 55.

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 programmable circuits can introduce security, stability, and performance risks due to the impact of the logic executed on the programmable circuits on other aspects of the system, particularly when the programmable logic must be updated or replaced while other users and virtual machines continue to operate. *Id.*

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.*

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 “the customer defined logic from communicating with other circuits of computing system 300.” *Id.* Importantly, access to the client configurable circuits is never prevented; they are always accessible. EX2009, ¶ 58. The shell only eases access to the client configurable circuits and determines what *other portions of the system* the client configurable circuits can access. *Id.*, ¶ 58. It does not determine whether the client configurable circuits can be accessed.

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 during reconfiguration, the patent enables seamless hardware modification without requiring virtual machines to be stopped. *Id.* Seshadri focuses on virtualization and control of programmable circuits rather than on authentication, locking, or routing.

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

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. Petitioners use Biederman to supply the “plurality of sandboxes” limitation in Grounds 2 and 4. Pet., 33, 68.

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

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 resources of the device. *Id.*, ¶ 59.

Central to Atsatt-113 is the architecture used to manage reconfigurable regions. Atsatt-113 contains a device controller 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, such as a user. *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.*

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.

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 authentication circuitry 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.*

Atsatt-113 also teaches other resources. 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 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.”). Petitioners allege that the “interconnection resources” are the claimed “bridge” but there is no ability to lock the “interconnection resources,” and they do not enable or enforce the masks.

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

Tarafdar teaches the use of management/performance-analytics tools to orchestrate virtual machines 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.

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

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.

**6. U.S. Patent Application Publication No. 2019/0041923A1
("Atsatt-923") (EX1009)**

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.

C. Alleged Unpatentability Grounds

Grounds	Claims	Statutory Basis	Prior Art
1	1-5, 10-14	§103	Seshadri
2	1-5, 9-14, 18	§103	Seshadri and Biderman
3	1-5, 10-14	§103	Atsatt-113
4	1-5, 9-14, 18	§103	Atsatt-113 and Biderman
5	6, 15	§103	Grounds 1-4 and Tarafdar
6	7, 16	§103	Grounds 1-4 and Kulkarni
7	8, 17	§103	Grounds 1-4 and Atsatt-923

III. PROSECUTION HISTORY

Petitioners discuss that the '943 patent was rejected in an Office Action. Pet., 11. However, as Petitioners concede, the Examiner found it persuasive when Applicant argued that the cited art did not disclose a PLD, which is “a term of art referring to a specific type of integrated circuit that has user programmable

electronic circuits.” *Id.* (citing EX1002, 000130–132.). Further, the Examiner issued the patent after Applicant added the locking/unlocking limitations, reflecting that these issues were considered when analyzing patentability during prosecution. *Id.* (citing EX1002, 000141-142)

IV. PRIORITY DATE OF THE '943 PATENT

Petitioners claim the '943 patent has a priority date of July 15, 2018. Pet., 12. Patent Owners do not dispute, for purposes of this Response, that all of Petitioners' prior art pre-dates the priority date of the '943 patent.

V. LEVEL OF ORDINARY SKILL

Petitioners identify their proposed level of ordinary skill. Pet., 14. For purposes of this proceeding, Patent Owners do not dispute Petitioners' proposed level of ordinary skill, nor do Patent Owners dispute that Petitioners' expert, Dr. Alpert, is a POSITA.

Patent Owners' expert, Dr. Paul Min, is also a POSITA because he has a Ph.D., M.S., and B.S. in Electrical Engineering. EX2009, ¶¶ 9-22 (citing EX2010). Dr. Min is currently a Senior Professor in the Department of Electrical Systems Engineering at Washington University, St. Louis, and has worked in the field of electrical engineering and data processing, streaming architecture, and data networks for more than thirty years. *Id.*

VI. RELATED MATTERS

Patent Owners, CV/XE, presently assert the ‘943 patent in the following district court action: *Concurrent Ventures, LLC and XtreamEdge, Inc. v. Advanced Micro Devices, Inc. and Pensando Systems, Inc.*, No. 4:25-cv-09567-JST (N.D. Cal. 2025).

Petitioners have two other instituted *inter partes* reviews on two other patents in the district court action: IPR2025-00478 and IPR2025-00486.

VII. CLAIM CONSTRUCTION

A. “programmable logic device”

The Petition identifies one term that is subject to dispute in the district court litigation: “programmable logic device.” Pet., 15. Petitioners argue that “programmable logic device” means “an IC where the user can alter the connection pattern of the circuits.” *Id.* Patent Owners do not agree with this construction and proposed a different construction in district court, however, Petitioners’ patentability theory fails even if their construction is applied. Thus, the term “programmable logic device” need not be construed in this proceeding. *See Promptu Sys. Corp. v. Comcast Corp.*, 92 F.4th 1372, 1380 (Fed. Cir. 2024) (“[O]nly those terms need be construed that are in controversy, and only to the extent necessary to resolve the controversy.” (quoting *Vivid Technologies, Inc. v. American Science & Engineering, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

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

The Petition ignores that the inventors acted as lexicographers in defining the meaning of “hardware-based” for purposes of the ’943 patent. This makes claim construction necessary. The term “hardware-based router” means a router that “lacks a processor.” The ’943 patent expressly defines a hardware-based “router with a port”:

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). In the ’943 patent, any “router with a port” that is executed by a processor is inherently not hardware. EX2009, ¶ 51. In contrast, a “router with a port” is hardware-based when it is executed directly, without a processor. *Id.*

Petitioners appear to base their theories on a broader and unsupported interpretation under which a “router with a port” is “hardware-based” as long as it is on a hardware device, such as 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 executed without a processor, simply because they are implemented on a hardware device. Pet, 24, 58. Such construction is inconsistent with the definition supplied by the ’943 patent.

EX1001, 5:1-8. Further, 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 a hardware device, such as an FPGA, is hardware-based, contrary to the explicit disclosure in the ’943 patent. *See* EX1001, 5:1-8, 6:20-27, 6:59-60, 7:15-20.

VIII. PETITIONERS HAVE NOT PROVEN THAT ANY CHALLENGED CLAIM IS UNPATENTABLE

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

Grounds 1 and 2 fail to render obvious claims 1 and 10 for two reasons. First, 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 components* within the device. There is no disclosure or suggestion that access to the client configurable circuits is ever prevented—and certainly not that access to the client configurable circuits is prevented by locking the bridge. EX2009, ¶ 71. Second, Seshadri does not disclose a router or port that is “hardware-based” (*i.e.*, that the “router with a 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.” Pet., 27. But that is not true—Seshadri’s FPGA includes processors, in addition to the “programmable fabric,” and Seshadri does not disclose that the identified “router with a port” is not executed in the processor. If anything, Seshadri suggests exactly that: the identified “router with a port” is executed by a processor, and thus not “hardware-based.”

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.”

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.”² EX1001, cl. 1. Petitioners allege that Seshadri’s shell logic locks or unlocks to prevent or enable user access to sandboxes, but Seshadri discloses no such functionality. Instead, Seshadri discloses only that the client configurable circuit access to other components of the peripheral device can be restricted. EX2009, ¶ 74.

² Petitioners’ arguments are based only on Seshadri. Pet., 20-31. Petitioners rely on Biederman only to argue that a system containing “a plurality of sandboxes” was known in the art. *Id.*, 33-34.

Seshadri does not disclose that access to the sandboxes is ever prevented, as required by each claim.

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 go on to assert that “[t]he bridge is lockable/unlockable to prevent or enable user access to each sandbox.” *Id.*, 29. But Petitioners’ assertion is made up of whole cloth and finds no support in the specification.³

³ Dr. Alpert’s declaration likewise adds nothing to Petitioners’ argument. Dr. Alpert’s testimony largely mirrors the Petition without providing additional support for these limitations and should thus be discounted. *See Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15-17 (PTAB Aug. 24, 2022) (designated precedential) (according “little weight” to declaration testimony that contains a verbatim restatement of a petition’s conclusory assertions without additional supporting evidence or reasoning).

To support their assertion, 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 the cited portions of Seshadri do not support Petitioners’ argument; Seshadri discloses no such thing. 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 isolates it “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. Seshadri is simply explaining that 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.

It makes sense that the client configurable circuits can only be accessed via the shell logic because one purpose of the shell logic is to make configuration of the circuits easier. 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. EX2009, ¶ 77.

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)).

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, Petitioners mischaracterize Seshadri to read limitations into it that do not exist.⁴ Seshadri makes only a single reference to “customer account permissions,”

⁴ 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. Petitioners’ footnote is of no moment. 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

and that sentence is fatally short on details. It reads in full: “First, the customer account permissions can be checked.” EX1004, 9:54. That single line fails 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.

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 baldly assert that the customer account permissions are checked to “reject or accept the request to unlock the reprogramming functionality,” Pet., 30, but they pull that conclusion out of thin air. 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. EX2009, ¶ 80. 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.” EX1004, 9:48-

logging in can lead to the bridge can be locked to prevent access to the sandboxes. Pet. 30, EX1021, 000034-35, 000055-56. Showing that the system accepts login credentials is not itself sufficient to establish unpatentability; logging in by itself does not disclose or suggest the claimed locking and unlocking.

53 (emphasis added). Seshadri explains that the *host* is vulnerable while new customer logic is executed on the client configurable circuits, because “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.

That Petitioners resort to misleading quotations shows the lack of actual support for their argument. 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 is saying is that after the old customer logic has been cleared out, the new customer logic can “subsequently” be loaded. EX2009, ¶ 82. There is no disclosure of locking the shell logic to prevent access to the client configurable circuits.

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. 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 portions of 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. EX2009, ¶ 83.

Finally, Petitioners 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 portion 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. Petitioners' assertion that it is locking and unlocking the bridge that achieves these goals is *ipse dixit* and hindsight.

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.").

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. As is apparent, 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 just 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, Dr. Min explains that 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. EX2009, ¶ 87.

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

Petitioners do not show that Seshadri contains a “hardware-based”⁵ “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,”⁶ (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*.⁷

⁵ Petitioners only argue that Seshadri discloses a “hardware-based” router with a port, not one that is “firmware-based.” Pet., 27.

⁶ The Board found that Petitioners’ alleged “router with a port” and “bridge” was the “shell logic” generally. ID, 21-22.

⁷ The Board in the Institution Decision determined that “[b]ased on the current record and for the purposes of institution, Seshadri sufficiently teaches limitation [1b]” because Seshadri discloses that the shell logic functionality is implemented in an FPGA, ID, 18-19. The Board, however, explicitly invited Patent Owners to propose a construction other than the plain and ordinary meaning should they believe one

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,” defeating their ground from the start. *See Trove Brands, LLC d/b/a BlenderBottle Co. v. CamelBak Prods., LLC*, IPR2024-01137, Paper 34 at 39 (P.T.A.B. Jan. 15, 2026) (argument fails where it applies the wrong claim construction).

Even had Petitioners alleged the shell logic is executed without a processor, Seshadri offers no support for such an allegation. FPGAs contain processors on which software can perform routing and port functionalities. EX2009, ¶ 90. Thus, while FPGAs do include hardware-based portions, they also have processors and software-based functionalities implemented with those processors. *Id.* The fact that the shell logic is on an FPGA does not mean that it is hardware-based. *Id.* Seshadri

appropriate. ID, 18 n.16. Patent Owners have proposed such a construction and explained why it should be adopted. *See supra* § VII.b. While Patent Owners continue to disagree with the Board’s analysis in the ID, it is not determinative of the question here: whether Seshadri discloses a “hardware-based” “router with a port” under the correct construction, which requires that the “router with a port” lacks a processor. The Board has not yet had an opportunity to reach that question.

does not disclose or suggest that any or all of the shell logic (Petitioners' alleged "router with a port") is executed without a processor.

If anything, 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. Crucially, it is not the shell logic itself that is programmed in the configurable circuits, but rather the shell logic "wraps" around the circuits to perform software operations. EX2009, ¶ 92. 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:

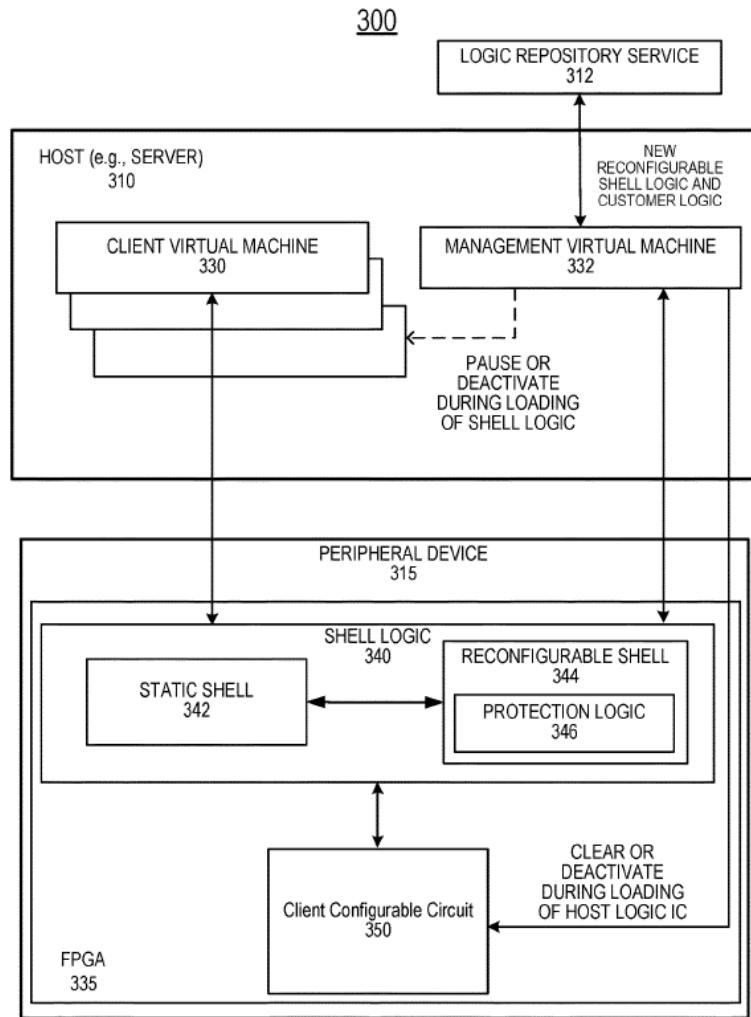


FIG. 3

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:

400

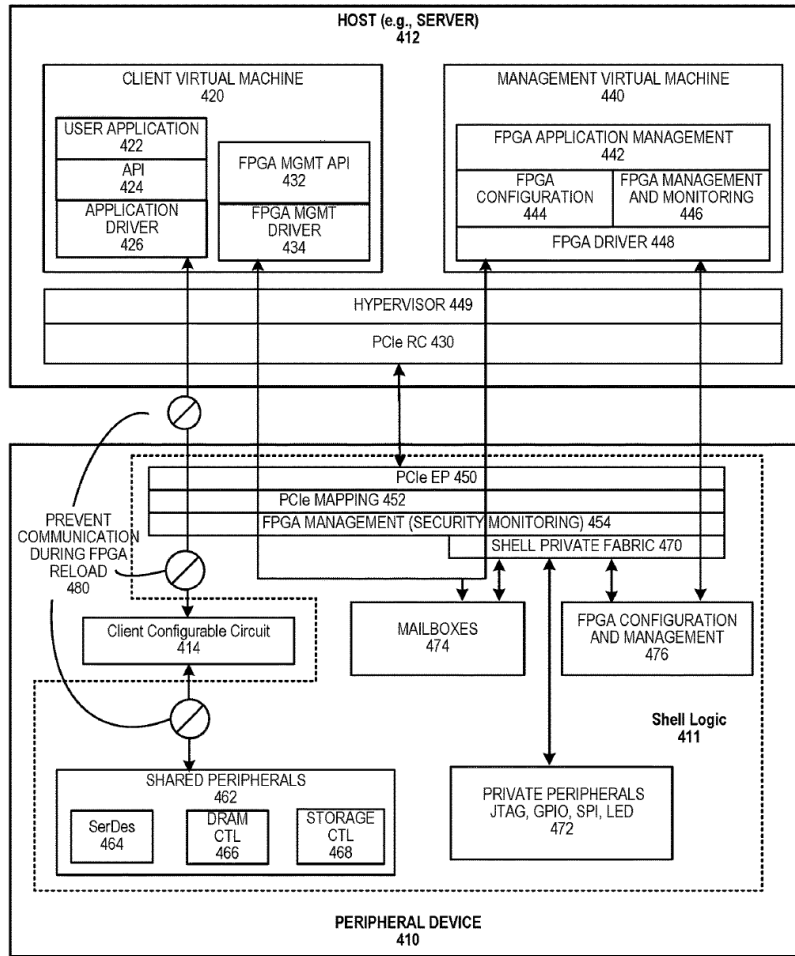


FIG. 4

Id., FIG. 4.

Petitioners 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.” *See In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364,

1380 (Fed. Cir. 2016) (petitioners cannot carry their burden with conclusory statements that are not based on evidence in the record).

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 that 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. EX2009, ¶ 94.

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 2:2-5 and 3:24-27 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 (emphasis added), 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.” (emphasis added)). The client-configurable circuit may be programmed “with” a shell logic. The shell logic thus accompanies the client-configurable circuit; it is separate from the client-configurable circuit. EX2009, ¶ 95. This is made even clearer in the figures cited above where the shell logic sits outside and separate from the client-configurable circuits.

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*. Dr. Min explains 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. EX2009, ¶ 96. 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. *Id.*

Petitioners’ expert testimony also does not show otherwise. Dr. Alpert parrots the Petition with 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. 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. EX2009, ¶ 97. A POSITA would recognize that in the context of the '943 patent, all executable logic, whether software, firmware, or hardware, ultimately runs on electronic circuits, and the relevant distinction is not whether circuits are involved with the “router with a port,” but whether the functionality is implemented directly in fixed hardware or expressed as instructions stored in memory and executed by a processor. *Id.* 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. *Id.*

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 actually assume that Seshadri’s alleged “port” is implemented in software because modern networking devices rely heavily on virtual ports. EX2009, ¶ 98. 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. *Id.* 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. *Id.* If the port were in hardware, a POSITA would expect Seshadri to disclose physical connections between the router and downstream sandboxes. *Id.* 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. EX2009, ¶ 98.

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 they cannot carry their burden in proving unpatentability without showing 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.”

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.

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.

Having failed to establish unpatentability through the combination of Seshadri and Biederman, Petitioners turn to alternative grounds based on Atsatt-113 and Biederman. These grounds also fail for multiple reasons. First, Atsatt-113 does not disclose or render obvious a bridge that is lockable or unlockable or that access to the alleged sandboxes can be prevented. Second, Atsatt-113 does not disclose or render obvious a "router with a port" in the "first region," as required by the claims.

Third, Atsatt-113 does not disclose or render obvious that the alleged “router with a port” is “hardware-based,” as properly construed. Petitioners have also not shown that the limitations of claims 4, 5, 13, and 14 are disclosed or rendered obvious.

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.”**

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. Petitioners map this locking/unlocking to Atsatt-113’s “masking” functionality, which provides each alleged sandbox (Atsatt-113’s partitions⁸) 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

⁸ Atsatt-113 explains that “partitions” are reconfigurable regions of the programmable fabric that group together resources of the programmable device. EX1006, ¶¶ 5, 39.

claimed “bridge” and enforce the masks and what resources each sandbox may access. Pet., 65-66.

Petitioners’ mapping suffers from two flaws. First, Atsatt-113’s “masks” do no prevent or allow access to the sandboxes. Crucially, users can *always access the 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.” Petitioners have thus failed to show that Atsatt-113 discloses or suggests a bridge that is lockable and unlockable to prevent or allow user access to sandboxes.

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

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 specifically 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. Atsatt-113, on the other hand, describes a very 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 a form of identification *for the sandboxes* that state what resources those sandboxes may access. 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 to the sandboxes without restriction, because it is only concerned with which resources the sandboxes themselves may access.

This difference is more than just form; it is substance. 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).

Petitioners’ citations to Atsatt-113 do not 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 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.

Petitioners’ own arguments show the mismatch between Atsatt-113 and the ’943 patent. While Petitioners are wrong that Atsatt-113 discloses using masks to prevent access to sandboxes, their 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. EX2009, ¶ 110. Atsatt-113 does not disclose or suggest any such scenario. Nor would such a scenario make any sense to a POSITA. 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. *Id.* Ultimately, Atsatt-113's masks are used to ensure that partitions do not change resources that other partitions are relying on—they are ill-suited for preventing access to all resources.

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 administrator gains centralized and immediate authority over user access without needing to modify individual user credentials. EX2009, ¶ 111. In such an architecture, the operator can allow or deny access at any time without altering the underlying user-mask ownership structure. *Id.* Ultimately, Atsatt-113 does not disclose or suggest anything close to the claimed arrangement.

ii. Atsatt-113's "Interconnection Resources" Cannot Be Locked or Unlocked and Do Not Enforce the "Masks."

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. But 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.”).

None of Petitioners’ evidence 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. EX2009, ¶ 114. 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.

Petitioners contend that 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 wholly absent

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. The interconnection resources, at most, act as a routing fabric, not an enforcement mechanism. EX2009, ¶ 116. 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 interconnection resources cannot be responsible for mask authentication.

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* are locked or unlocked. EX2009, ¶ 117. 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. EX1006, ¶¶ 79-81.

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.”

Petitioners have not shown that Atsatt-113 discloses 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.” Pet., 59, 61. But the global resources are only resources shared between the partitions in the *second region*, not the first region. EX1006, ¶ 55. Petitioners’ have thus failed to show Atsatt-113 discloses a “router with a port” in the first region, as required by the claim.

Petitioners are wrong that Atsatt-113's global resources are in the first region.

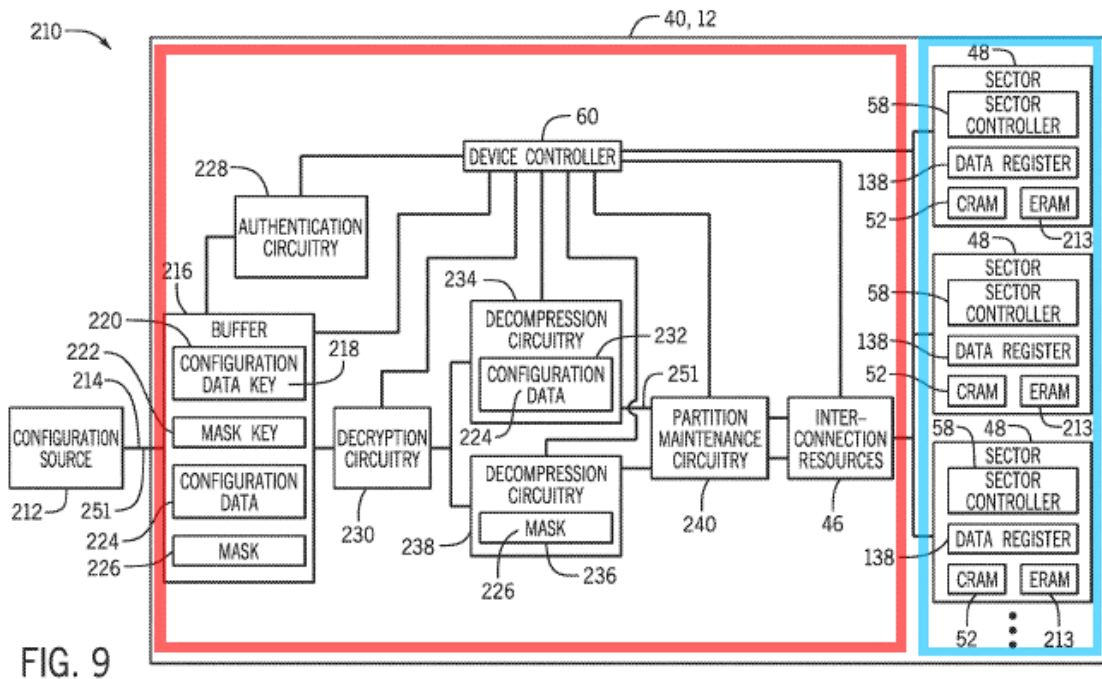
Atsatt-113 explains that global resources are resources, such as input/output ports and serial interfaces, that are shared *between the partitions in the sectors* comprising the second region of the system:

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. EX2009, ¶ 120.

Dr. Alpert's opinion confirms this understanding. As an initial matter, Dr. Alpert 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).



FIG. 3

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

Petitioners' mapping is inconsistent 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. EX2009, ¶ 122. Petitioners, however, present an analysis in which these regions are indiscriminately grouped together within the second region. Pet., 58-64.

Petitioners cite to some functionality performed by the “device controller 60” and “interconnection resources” to show that the “global resources” maps to the claimed “router with a port.” Pet., 58. But Atsatt-113 is clear that the device controller and interconnection resources are *not* part of the global resources. 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. 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). Petitioners cannot meet their burden of proving unpatentability with this contradictory, scattershot evidence.

Petitioners cannot meet their burden in proving unpatentability because limitations 1[b] and 10[b] require the router be in the first region of the system. EX1001, cl. 1, 10. Atsatt-113 makes clear that the global resources are not in the first region, they are in the second region.

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

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).

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, which a POSITA would understand means a processor (*i.e.*, not hardware-based). EX1006, ¶ 55; EX2009, ¶ 126.

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 processors or engines. EX2009, ¶ 127.

2. Claims 4 and 13

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—a POSITA would not view configuring the system at setup to mean the bridge is “lockable” *and* “unlockable,” as required by the claims. EX2009, ¶ 128.

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. EX2009, ¶ 129. Because global resources are not in the first region, the base partition within those global resources cannot be the lockable bridge and router.

Setting aside Petitioners’ failure to identify any bridge or router, 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 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.

Petitioners further argue that the base partition is “lockable” because it “is locked after initial setup.” Pet. 76. But as Dr. Min explains, a device being defined or configured at setup does not mean that it is lockable, it simply means that initialization has occurred. EX2009, ¶ 131. A lockable device, by definition, is one that is both capable of being locked and unlocked. *Id.* Merely establishing a device’s configuration at setup does not demonstrate that its state can later be changed. *Id.* And it certainly 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).

Nor 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. EX2009, ¶ 132. 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. *Id.* 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. *Id.*

3. Claims 5 and 14

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

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 hash 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.

As discussed above, the ’943 patent describes that the bridge has a security layer designed to protect system circuitry, including by authenticating users. EX1001, 5:43-45. But in Atsatt-113, the only evidence that Petitioners provide shows that 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*. EX2009, ¶ 134. 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, to simply authenticate the data as the same data that was sent by the user. EX2009, ¶ 134. In sum, Atsatt-113 describes using authentication keys to confirm the data that it has received is the correct data, not to authenticate a user or verify a permission to reconfigure one or more sandboxes.

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

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. Tarafdar thus does not address the deficiencies in Grounds 1–4.

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

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. Kulkarni thus does not address the deficiencies in Grounds 1–4.

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

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. Atsatt-923 thus does not address the deficiencies in Grounds 1–4.

IX. CONCLUSION

For the reasons stated above, the Board should find that Petitioners failed to carry their burden of proving unpatentability as to all grounds in the Petition.

Dated: January 29, 2026

By: /s/ Michael Matulewicz-Crowley
Michael Matulewicz-Crowley
PTO Reg. No. 72,220
Counsel for Patent Owner

CERTIFICATE OF COMPLIANCE

The undersigned hereby certifies that Patent Owners' Response (the "POR") consists of 11,578 words, excluding the table of contents, certificate of service, this certificate of compliance, or table of exhibits. The POR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare this paper (Microsoft Word).

Dated: January 29, 2026

/s/ Michael Matulewicz-Crowley

Michael Matulewicz-Crowley
Counsel for Patent Owner

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6, the undersigned certifies that on January 29, 2026, a copy of the foregoing document was served by email upon the following:

Harper Batts (Reg. No. 56,160)
hbatts@sheppardmullin.com
legaltm-amd-concurrent@sheppardmullin.com
SHEPPARD MULLIN LLP
1540 El Camino Real, Suite 120
Menlo Park, CA 94025
T 650-815-2673 / F 650-815-4668

Christopher Ponder (Reg. No. 77,167)
cponder@sheppardmullin.com
legaltm-amd-concurrent@sheppardmullin.com
SHEPPARD MULLIN LLP
1540 El Camino Real, Suite 120
Menlo Park, CA 94025
T 650-815-2676 / F 650-815-4664

Jeffrey Liang (Reg. No. 69,043)
jliang@sheppardmullin.com
legaltm-amd-concurrent@sheppardmullin.com
SHEPPARD MULLIN LLP
1540 El Camino Real, Suite 120
Menlo Park, CA 94025
T 650-815-2685 / F 650-815-4666

/s/ Michael Matulewicz-Crowley
Michael Matulewicz-Crowley
Counsel for Patent Owner

CLAIM APPENDIX

The language of independent claims 1 and 4 of the '943 patent is reproduced below. For convenience, Patent Owners have included the limitation labels used by Petitioners.

Claim 1:

[1pre]. A programmable device for data flow processing in a user-configurable server with swappable pods or cards, comprising:

[1a] a programmable logic device (PLD) having a first region and a second region;

[1b] the first region comprising a hardware-based or firmware-based router with a port enabling communication between the router and the second region; and

[1c] the second region comprising one or more sandboxes with user-definable programmable electronic circuits of the PLD,

[1d] wherein the port comprises: a bridge being lockable to prevent user access, and unlockable to enable user access, to each of the sandboxes and the user-definable programmable electronic circuits therein.

EX1001, cl. 1; *see also* Pet., 100 (claim appendix).

Claim 4:

The programmable device of claim 1, wherein the port comprises:

a bridge being lockable to prevent reconfiguration of the bridge and prevent reconfiguration of the router while enabling reconfiguration of the one or more sandboxes.

EX1001, cl. 4; *see also* Pet., 100–01 (claim appendix).

Claim 5:

The programmable device of claim 1, wherein the port comprises:

a bridge having a security layer to authenticate a user or verify a permission, to reconfigure one or more of the one or more sandboxes.

EX1001, cl. 5; *see also* Pet., 101 (claim appendix).