IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

CONCURRENT VENTURES, LLC AND XTREAMEDGE, INC., Plaintiffs, v. ADVANCED MICRO DEVICES, INC. AND PENSANDO SYSTEMS, INC., Defendants.	Civil Action No. 1:24-cv-335-DII

Defendants Advanced Micro Devices, Inc.'s and Pensando Systems, Inc.'s Invalidity Contentions

Pursuant to the Schedule Order (Dkt. 52), Defendants Advanced Micro Devices, Inc. and Pensando Systems, Inc. hereby disclose their Invalidity Contentions.

The Invalidity Contentions set forth below are based on information currently known to Defendants. Discovery in this case is still in its early stages, and Defendants' investigation and analysis of Plaintiffs' claims and Defendants' defenses are ongoing. Defendants also have not yet obtained discovery from the named inventors identified on the Asserted Patents or the individuals involved in prosecuting the respective applications leading to the Asserted Patents, nor have Defendants completed discovery from third parties that may have evidence relating to prior art. Defendants therefore reserve the right to supplement or modify these Invalidity Contentions based on further discovery, evaluation of the scope and content of the prior art, or changes in any of Plaintiffs' asserted claims or infringement contentions.

The Court has not yet issued any claim-construction rulings. Defendants' Invalidity Contentions herein are based on the claim constructions apparently underlying Plaintiffs' infringement contentions. Defendants' Invalidity Contentions are not intended to, and do not,

reflect any acquiescence or concession by Defendants as to the proper scope or construction of the asserted claims. To the extent that these Invalidity Contentions reflect any claim construction consistent with Defendants' infringement contentions, no inference is intended, nor should any be drawn, that Defendants agree with Plaintiffs' claim constructions, and Defendants expressly reserve the right to contest such constructions. Moreover, Defendants' Invalidity Contentions should not in any way be read as an admission that any of their products or technology infringes any claim of any Asserted Patents; Defendants specifically deny such infringement.

The prior-art references identified herein may disclose the limitations of the Asserted Claims either explicitly or inherently, or may be relied upon to show the state of the art in the relevant time frame. The discussion of the different references herein and in the attached claim charts may reflect alternative applications of the prior art against the Asserted Claims—comments on one item of prior art should not be construed as necessarily applying to any of the other prior art discussed herein. For example, the suggested obviousness combinations are provided in the alternative to Defendants' anticipation contentions and are not to be construed to suggest that any reference included in the combinations is not by itself anticipatory. To the extent that Plaintiffs contend that limitations of the Asserted Claims are not disclosed in the anticipatory prior art identified herein, Defendants reserve the right to identify other references and combinations that address those arguments, and reserve the right to so modify, amend, or supplement these Invalidity Contentions.

Defendants' citations to specific portions of the prior art references relied upon herein should not be construed to mean that other portions of the prior art references are not relevant to the invalidity of the Asserted Claims. Defendants reserve the right to rely on the entirety of any

or all of the prior art references (whether or not claim-charted) as a basis for asserting invalidity of the Asserted Patents, and/or as necessary to supplement their contentions with additional citations and evidence.

I. Claim Charts Setting Forth Where in the Prior Art Each Element of the Asserted Claims Are Found

In the below subsections, Defendants identify each item of prior art that anticipates and/or renders obvious each Asserted Claim. The prior art identified herein also includes items that may be relied upon to show knowledge or use by others under 35 U.S.C. § 102(a)(1), public use or on-sale bar under § 102(a)(1), prior filing under § 102(a)(2), derivation or prior inventorship under §§ 135 and 291, and the identity of any person or entity involved in and the circumstances surrounding the making of the invention before the application for each Asserted Patent, including relevant dates where presently known. The prior art cited and discussed below and in the attached claim charts anticipates and/or renders obvious, alone or in combination with the below-disclosed references, one or more of the Asserted Claims.

Defendants have provided limitation-by-limitation claim charts demonstrating where each limitation of the Asserted Claims of the Asserted Patents is found in the prior art. These claim charts are based on Defendants current understanding of the terms and limitations of Plaintiffs' Asserted Claims, and Defendants' current understanding of the claim constructions underlying Plaintiffs' infringement contentions. To the extent any claim limitation is deemed not to be disclosed or otherwise satisfied by a given item of prior art, Defendants contend that the difference, if any, between the claim and such prior art would have been within the knowledge of and obvious to one of ordinary skill in the relevant art at the time of the alleged invention, such that the claimed invention would have been obvious both in light of the prior-art item alone and/or in light of a combination of prior art.

Defendants have provided exemplary citations to each prior-art item relied upon for each limitation of each Asserted Claim. These citations represent neither every location where a particular claim limitation is found in the prior art nor every instance in which such prior art satisfies the claim limitation. Rather, Defendants reserve the right to rely on additional, or different, portions or aspects of the prior art other than those specifically cited in the claim charts provided herewith. Moreover, Defendants may rely on other portions or aspects of the prior art as aids to understanding and interpreting the portions that are cited. Where Defendants cite a particular figure in a prior-art reference, such citation should be understood to encompass the caption and description of the figure and any text relating to the figure, in addition to the figure itself. Conversely, where a cited portion of text refers to a figure, the citation should be understood to include the figure as well. Defendants may additionally rely on information incorporated by reference in the identified prior art to provide additional support for Defendants' contention for a given claim limitation.

Defendants reserve the right to supplement and/or amend these contentions based on the herein-identified prior art considered individually or in combination with additional information learned during discovery and/or in light of claim-construction developments in this case. Defendants may also become aware of additional prior art as its investigation and discovery in this case progress, and Defendants reserve the right to use any such prior art to demonstrate that the Asserted Claims of the Asserted Patents are invalid.

Defendants contend that the prior-art references identified below are anticipatory references under 35 U.S.C. §§ 102(a)(1)-(2). To the extent any of the anticipatory references were embodied in actual systems, such systems may constitute prior art based upon which

Defendants will continue to develop further invalidity defenses as discovery and Defendants' investigation progress.

As the Supreme Court has established, "[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results." *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 416 (2007).

> When a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If one of ordinary skill can implement a predictable variation, § 103 likely bars its patentability. For the same reason, if a technique has been used to improve one device, and one of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

Id. at 417. "[I]n many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.* at 420. Indeed, it is sufficient that a combination of elements was "obvious to try":

> When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.

Id. at 421. "In that instance the fact that a combination was obvious to try might show that it was obvious under § 103." *Id.* The Supreme Court has admonished against the application of "[r]igid preventative rules that deny factfinders recourse to common sense…." *Id.* This is because "[g]ranting patent protection to advances that would occur in the ordinary course without real innovation retards progress and may, in the case of patents combining previously known elements, deprive prior inventions of their value or utility." *Id.* at 419.

For each Asserted Patent, all prior-art items relied upon herein are directed to the same fields of endeavor. Accordingly, one of ordinary skill in the art would have recognized that the results of the combinations were predictable, and would have been motivated to modify and/or

combine the prior-art items for each Asserted Patent to arrive at the claimed inventions. At least the following rationales support a finding of obviousness based on such combinations of prior art:

A. combining prior-art elements according to known methods to yield predictable results;

B. simple substitution of one known element for another to obtain predictable results;

C. use of known techniques to improve similar devices (or methods) in the same way;

D. applying a known technique to a known device (or method) ready for improvement to yield predictable results;

E. choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success ("obvious to try");

F. prompting of variations of known work in one field of endeavor for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one skilled in the art; and

G. some teaching, suggestion, or motivation in the prior art that would have led one skilled in the art to modify the prior art or to combine prior-art teachings to arrive at the claimed invention.

Thus, to the extent that the Asserted Claims are deemed not to be anticipated, Defendants contend that the Asserted Claims are invalid as obvious under 35 U.S.C. § 103 in view of at least the prior art listed herein and set forth in the attached Exhibits. Defendants' contentions that the Asserted Claims are obvious under 35 U.S.C. § 103 are in no way an admission or suggestion that any prior art relied upon herein is not anticipatory under 35 U.S.C. § 102.

Defendants set forth certain specific exemplar combinations of prior art below in addition to combinations identified in its claim charts, based on Defendants' present understanding of the Asserted Claims and the apparent claim constructions adopted by Plaintiffs. As discussed above,

Defendants contend that each prior-art item identified above is anticipatory under 35 U.S.C. § 102 and/or alone renders the claimed invention obvious under 35 U.S.C. § 103.

A. U.S. Patent No. 8,924,596

The following table identifies the prior art for the '596 patent. Each item below constitutes prior art under one or more provisions of 35 U.S.C. §§ 102(a)(1)-(2) by reason of its status as a publication and/or patent disclosing, or a system on sale and/or in public knowledge or use in the United States embodying the claimed invention before Plaintiffs' asserted priority date of the '596 patent. For each item of prior art, Defendants have attached a claim chart identifying where each claim limitation is found in the cited prior art.

Table A: Prior Art to the '596 Patent

Prior Art to '596	Anticipation/ Obviousness	Claim Chart
U.S. Patent Application Publication No. 2011/0161976 to Alexander et al. ("Alexander"); filed on December 30, 2009; published on June 30, 2011.	A & O	Exhibit 596-01
U.S. Patent No. 6,229,813 to Buchko et al. ("Buchko"); filed on November 25, 1998; issued on May 8, 2001.	A & O	Exhibit 596-02
U.S. Patent No. 7,743,197 to Chavan et al., ("Chavan"); filed on May 11, 2006; issued on June 22, 2010.	A & O	Exhibit 596-03
U.S. Patent No. 8,559,439 to Dongare et al. ("Dongare"); filed on November 3, 2011; issued on October 15, 2013.	O	Exhibit 596-04
U.S. Patent No. 6,173,307 to Drews ("Drews"); filed on August 20, 1998; issued on January 9, 2001.	O	Exhibit 596-05
U.S. Patent No. 8,051,227 to Gewirtz et al. ("Gewirtz"); filed on May 10, 2010; issued on November 1, 2011.	O	Exhibit 596-06

Prior Art to ’596	Anticipation/ Obviousness	Claim Chart
U.S. Patent No. 6,055,579 to Goyal et al. (“Goyal”); filed on November 17, 1997; issued on April 25, 2000.	O	Exhibit 596-07
U.S. Patent No. 5,204,954 to Hammer et al. (“Hammer”); filed on July 29, 1992; issued on April 20, 1993.	A & O	Exhibit 596-08
U.S. Patent No. 7,206,901 to Kanai et al. (“Kanai”); filed on April 1, 2004; issued on April 17, 2007.	O	Exhibit 596-09
U.S. Patent No. 7,398,528 to Khawand et al. (“Khawand”); filed on November 13, 2004; issued on July 8, 2008.	A & O	Exhibit 596-10
U.S. Patent Application Publication No. 2003/0149820 to Kolinummi et al. (“Kolinummi”); filed on January 15, 2003; published on August 7, 2003.	O	Exhibit 596-11
U.S. Patent No. 8,463,958 to Mannava et al. (“Mannava”); filed on August 8, 2011; issued on June 11, 2013.	O	Exhibit 596-12
U.S. Patent No. 6,810,462 to Matsunami et al. (“Matsunami”); filed on August 19, 2002; issued on October 26, 2004.	O	Exhibit 596-13
U.S. Patent No. 7,865,684 to Michaylov et al. (“Michaylov”); filed on June 27, 2005; issued on January 4, 2011.	O	Exhibit 596-14
U.S. Patent No. 8,028,284 to Ohkawa (“Ohkawa”); filed on October 31, 2006; issued on September 27, 2011.	O	Exhibit 596-15
U.S. Patent No. 5,805,930 to Rosenthal et al. (“Rosenthal”); filed on March 28, 1997; issued on September 8, 1998.	O	Exhibit 596-16
U.S. Patent No. 5,907,485 to Van Loo et al. (“Van Loo”); filed on March 31, 1995; issued on May 25, 1999.	A & O	Exhibit 596-17
ARM System(s)	A & O	Exhibit 596-18

Prior Art to ’596	Anticipation/ Obviousness	Claim Chart
Intel/Fulcrum System(s)	A & O	Exhibit 596-19
Nvidia/Mellanox System(s)	A & O	Exhibit 596-20

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the Asserted Claims of the ’596 patent, including for the reasons illustrated in the accompanying claim chart and based on Plaintiffs’ infringement allegations and contentions for the ’596 patent.

2. Obviousness

A person of ordinary skill in the art would have been motivated to combine any reference in Table A with any other reference(s) in Table A. To the extent that any prior art identified in Table A above is deemed to not anticipate an Asserted Claim, the prior art in Table A may be combined with one or more prior art references in Table A to render the claim obvious. All of these items of prior art are directed to and propose solutions similar to those discussed in the ’596 patent, and is in the same field. All of these items of prior art are directed to multiprocessor systems and/or computer architecture and propose solutions similar to those discussed in the ’596 patent.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **Alexander + Drews**
- **Alexander + Gewirtz**
- **Alexander + Goyal**
- **Alexander + Kolinummi**
- **Alexander + Ohkawa**
- **Alexander + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Alexander discloses a multi-node processing system with queues, including a global command queue. *See* Abstract. Alexander describes its technical field as "multiprocessor data processing systems" and discloses a "method and system for completing parallel processing of work items of a single work set distributed across multiple processing units of a multi-node data processing system." ¶ [0002]. Alexander teaches a system with distributed processing across multiple processing units that require queue management and flow control at both the global and local level. ¶¶ [0044], [0049].

Therefore, given the context provided by Alexander, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Alexander with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Alexander with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine

Alexander with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Alexander with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Alexander with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Alexander with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Alexander because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Alexander discloses support for I/O operations. Therefore, a POSITA would naturally have looked to references that support and provide features for various I/O operations, including known storage systems. For example, a POSITA would have been motivated to combine Alexander with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Buchko + Drews**
- **Buchko + Gewirtz**
- **Buchko + Goyal**
- **Buchko + Kolinummi**
- **Buchko + Ohkawa**
- **Buchko + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Buchko discloses a multiprocessor system with distributed work among processors that share a queue. *See* Abstract. Buchko describes its technical field as "digital systems wherein processed items are temporarily stored in queues, typically having limited capacity, pending further processing and more particularly to a method and mechanism for controlling queue size in such systems." 1:6-12. Buchko explains that "there is a requirement for an improved mechanism for controlling the queue length or queue size in a multi task processor system or a multiprocessor system in which items removed from queues continue to use up system resources during subsequent processing." 2:36-40. Buchko acknowledges the need for flow control: "It is then placed in one of the aforementioned COS queues after the queue has been checked to determine whether there is sufficient room. This is done by checking the queue counter or calculating the space between the enqueue pointer and the virtual dequeue pointer. If there is not sufficient space the packet is discarded or one of the queued packets is discarded to make room for the new packet, if appropriate." 6:43-48.

Therefore, given the context provided by Buchko, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Buchko with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Buchko with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Buchko with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Buchko with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Buchko with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Buchko with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Buchko because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Buchko discloses the support of network operations. A POSITA would naturally have looked to references that support and provide features for various network operations, including network-based storage systems. For example, a POSITA would have been

motivated to combine Buchko with Kanai or Matsunami, as both teach multiprocessor-based networked-storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Chavan + Drews**
- **Chavan + Gewirtz**
- **Chavan + Goyal**
- **Chavan + Kolinummi**
- **Chavan + Ohkawa**
- **Chavan + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Chavan discloses an intelligent network processor with multiple processors. *See* Abstract, 1:35-37. Chavan describes its technical field as "Intelligent Network Processors (INPs) for use in computer network systems." 1:13-14. Chavan explains that its controller "creates queues in the external DDR memory for storing data messages, and also writes to and reads from such queues." 3:59-62. Chavan also acknowledges the need for flow control and teaches a credit-based system: "Credits are essentially the number of free buffers at a particular port for storing frames or packets." 6:54-55.

Therefore, given the context provided by Chavan, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA

would have been motivated to combine Chavan with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Chavan with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Chavan with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Chavan with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Chavan with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Chavan with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Chavan because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Chavan discloses the support of I/O operations and network-based storage such as SAN. A POSITA would naturally have looked to references that support and provide features for various I/O operations and network-based storage, including storage systems. For example, a POSITA would have been motivated to combine Chavan with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN

and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Dongare + Drews**
- **Dongare + Gewirtz**
- **Dongare + Goyal**
- **Dongare + Kolinummi**
- **Dongare + Ohkawa**
- **Dongare + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Dongare discloses a multiprocessor with multiple engines, input queues, and output queues. *See* Abstract. Dongare describes its technical field as "multi-engine processing environments. More particularly, the present disclosure relates to queue ordering in a multi-engine processing system." 1:14-17. Dongare explains that "commands within a command input queue are related only in their sequence and are otherwise independent of other commands in the input queue. Hence, the commands within a particular command input queue can be processed independent of each other." 5:27-32. Dongare also acknowledges the need for flow control: "The depth of the command output buffers can be chosen based on the bandwidth required by the system. The depth of a command output buffer refers to the number of processed commands or segments that can be stored in the command output buffer. By having a dedicated command buffer for each engine, each engine can directly access its command output buffer to write out processed commands without any waiting time." 12:20-27.

Therefore, given the context provided by Dongare, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Dongare with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Dongare with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Dongare with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Dongare with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Dongare with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Dongare with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Dongare because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Dongare discloses the support of I/O operations, including disk-based storage. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been

motivated to combine Dongare with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Hammer + Drews**
- **Hammer + Gewirtz**
- **Hammer + Goyal**
- **Hammer + Kolinummi**
- **Hammer + Ohkawa**
- **Hammer + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Hammer discloses a multiprocessor I/O unit. *See* Abstract. Hammer describes its technical field as "control of work flow between processes on a bus." 1:27-31. Hammer's system is a multiprocessor I/O system that consists of "peer processors," which causes "each processor [to] serve as many work requests as it generates work requests." 2:9-12. Hammer explains that the "bus hardware also controls queueing of messages from the bus units in main storage queues." Hammer also acknowledges the need for flow control: "Messages are transferred between bus managers to control and re-synchronize messages when a connection group does not have sufficient resources to service a request intended for a process connected to that group. When a target bus unit cannot accept additional work for a particular connection group, it returns a message with QUEUE FULL status specified." 10:59-65.

Therefore, given the context provided by Hammer, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Hammer with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Hammer with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Hammer with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Hammer with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Hammer with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Hammer with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Hammer because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Hammer discloses the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine

Hammer with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Khawand + Drews**
- **Khawand + Gewirtz**
- **Khawand + Goyal**
- **Khawand + Kolinummi**
- **Khawand + Ohkawa**
- **Khawand + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Khawand discloses a multiprocessor system that delegates tasks among its processors through the use of command task queues. *See* Abstract. Khawand describes its technical field as "mobile devices including an efficient multiprocessor system and methods thereof." 1:8-10. Khawand explains that "[e]ach task [] comprises at least one queue [] for processing at least one service application." 3:54-58. Khawand also acknowledges the need for flow control: "In step 604, the queue [] of the second processor [] is evaluated according to flow control criteria. … the flow control criteria can include a capacity threshold for each queue []." 4:6-16.

Therefore, given the context provided by Khawand, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Khawand with Drews, which teaches a queue reservation

system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Khawand with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Khawand with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Khawand with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Khawand with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Khawand with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Khawand because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Khawand discloses the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Khawand with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Mannava + Drews**
- **Mannava + Gewirtz**
- **Mannava + Goyal**
- **Mannava + Kolinummi**
- **Mannava + Ohkawa**
- **Mannava + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

One of ordinary skill in the art would have been motivated to combine Mannava with any other reference(s) in Table A. In addition to the above-mentioned motivations to combine, Mannava discloses a multiprocessor credit-based data processing system. *See* Abstract; Figure 3. Mannava describes its technical field as "data processing systems, and in particular, to systems having a number of initiator devices issuing transaction requests to a number of recipient devices and to the allocation of resources within the recipient devices for executing these transaction requests." 1:8-12. Mannava explains that "[r]ecipient device [] comprises a buffer [] for storing pending transaction requests that have been received prior to them being executed by the recipient device." 8:24-27. Mannava acknowledges the need for flow control: "Credit generator [] monitors to determine when there are available storage locations." 8:66-67.

Therefore, given the context provided by Mannava, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Mannava with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA

would have been motivated to combine Mannava with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Mannava with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Mannava with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Mannava with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Mannava with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Mannava because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Mannava discloses the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Mannava with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Michaylov + Drews**
- **Michaylov + Gewirtz**
- **Michaylov + Goyal**
- **Michaylov + Kolinummi**
- **Michaylov + Ohkawa**
- **Michaylov + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

One of ordinary skill in the art would have been motivated to combine Michaylov with any other reference(s) in Table A. In addition to the above-mentioned motivations to combine, Michaylov discloses a messaging queue system for a multiprocessor system. *See* Abstract. Michaylov describes its technical field as "managing message queues. Message queues can be used to provide asynchronous communication protocol for access entities (e.g., servers, operating systems, software modules, etc.) to exchange messages." 1:5-8. Michaylov explains that the "queue manager [] interacts with an input/output (I/O) manager [] that manages memory storage for a set of message queues with corresponding queue data structures." 3:24-31. Michaylov acknowledges the need for flow control: "determining if a space is available in each queue for writing a corresponding portion of the data, and if available, reserving the spaces in the queue." 1:25-29.

Therefore, given the context provided by Michaylov, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Michaylov with Drews, which teaches a queue

reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Michaylov with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Michaylov with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Michaylov with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Michaylov with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Michaylov with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Michaylov because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Michaylov discloses the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Michaylov with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Van Loo + Drews**
- **Van Loo + Gewirtz**
- **Van Loo + Goyal**
- **Van Loo + Kolinummi**
- **Van Loo + Ohkawa**
- **Van Loo + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Van Loo discloses a flow control method for uniprocessor and multiprocessor systems. *See* Abstract. Van Loo describes its technical field as "a new method and apparatus for flow control in a packet-switched microprocessor-based computer system (uniprocessor), a network of such computer systems, or a multiprocessor system. Van Loo explains that its system "classifies system interconnect queues as request queues, read-data queues, and write-data queues, and determines rules of transfer of both requests and data as loosely coupled events." Van Loo acknowledges the need for flow control: "The elimination of (advance or overflow) feedback signals in the present flow control system reduces the interface latency, since there is no extra handshake, no rescheduling or rearbitrating for resources, and no retries by the master." 3:38-41.

Therefore, given the context provided by Van Loo, a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Van Loo with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA

would have been motivated to combine Van Loo with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Van Loo with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Van Loo with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Van Loo with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Van Loo with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Van Loo because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Van Loo discloses the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Van Loo with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **ARM System(s) + Drews**
- **ARM System(s) + Gewirtz**
- **ARM System(s) + Goyal**
- **ARM System(s) + Kolinummi**
- **ARM System(s) + Ohkawa**
- **ARM System(s) + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, the ARM System(s) disclose message queues and other queues for buffering data. *See* Exhibit 596-18. For example, on information and belief, the ARM System(s) contain queues and support multiprocessors and semaphores through, for example, the use of "load register exclusive" (LDREX) and "store register exclusive" (STREX) instructions. *See id.* ARM microprocessors can be notified of queue status. *See id.*

Therefore, given the context provided by ARM System(s), a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine ARM System(s) with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Goyal, which teaches queue management based on a

centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with ARM System(s) because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, ARM System(s) disclose the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine ARM System(s) with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for many years to provide data storage over networks.

- **Intel/Fulcrum System(s) + Drews**
- **Intel/Fulcrum System(s) + Gewirtz**
- **Intel/Fulcrum System(s) + Goyal**
- **Intel/Fulcrum System(s) + Kolinummi**
- **Intel/Fulcrum System(s) + Ohkawa**
- **Intel/Fulcrum System(s) + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, the Intel/Fulcrum System(s) disclose receive and transmit queues for scheduling packets and congestion management based on status of the queues. *See* Exhibit 596-19. For example, the Intel/Fulcrum System(s) include a switch with a shared memory that is accessible to multiple ports, and it discloses maintaining receive and transmit queues for scheduling packets for transmission. *See id.* The Intel/Fulcrum System(s) include congestion management that maintains information about the sizes of different queues and determine if congestion occurs based on upper-bound limits on the queue sizes. *See id.* When the status of a queue is changed, notifications regarding the status of the queue is sent to one or multiple devices. *See id.*

Therefore, given the context provided by Intel/Fulcrum System(s), a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Drews, which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s)

with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Intel/Fulcrum System(s) because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Intel/Fulcrum System(s) disclose the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known and used in networked settings and data centers for years to provide data storage over networks.

- **Nvidia/Mellanox System(s) + Drews**
- **Nvidia/Mellanox System(s) + Gewirtz**
- **Nvidia/Mellanox System(s) + Goyal**
- **Nvidia/Mellanox System(s) + Kolinummi**
- **Nvidia/Mellanox System(s) + Ohkawa**
- **Nvidia/Mellanox System(s) + Rosenthal**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

In addition to the above-mentioned motivations to combine, Nvidia/Mellanox System(s) disclose maintaining queues and keeping track of the status of queues in the context of networking. *See* Exhibit 596-20. For example, Nvidia/Mellanox System(s) disclose a network architecture for communication among network adapters and I/O units attached to a switched fabric, where remote direct memory access is supported. *See id.* Queue pairs, including a send queue and a receive queue, are created and used for transferring data between nodes of a network. *See id.* The receive queue may be shared among different nodes. *See id.* The Nvidia/Mellanox System(s) further teach using a completion queue to track the status of send and receive queues and using a completion channel to notify nodes as to such status. *See id.* The status of a queue pair includes, for example, whether the receive queue is ready to receive, based on which the network of nodes operate. *See id.*

Therefore, given the context provided by Nvidia/Mellanox System(s), a POSITA would have naturally looked to other references that provide details related to multiprocessor systems, computer architecture, queue management, reservation systems, and flow control. For example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Drews,

which teaches a queue reservation system based on a counter for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Gewirtz, which teaches a programmable queue structure for multiprocessor systems with flow control based on a queue state register. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Goyal, which teaches queue management based on a centralized command processor for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Kolinummi, which teaches a reservation system using a hardware-based semaphore for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Ohkawa, which teaches queue management, queue structure, and queue placement for multiprocessor systems. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Rosenthal, which teaches queue management and flow control based on a flow control register that indicates the amount of space available in the queue for multiprocessor systems. A POSITA would naturally have looked to these references and been motivated to combine them with Nvidia/Mellanox System(s) because, e.g., they provide implementation details and facilitate resource sharing, optimize queue and reservation management, and reduce coordination overhead for accessing shared resources in multiprocessor systems.

Additionally, Nvidia/Mellanox System(s) disclose the support of I/O operations. A POSITA would naturally have looked to references that support and provide features for various I/O operations, including storage systems. For example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Kanai or Matsunami, as both teach multiprocessor-based storage systems, including support for NAS and SAN. SAN and NAS had been known

and used in networked settings and data centers for many years to provide data storage over networks.

B. U.S. Patent No. 9,529,767

The following table identifies the prior art for the ’767 patent. Each item below constitutes prior art under one or more provisions of 35 U.S.C. §§ 102(a)(1)-(2) by reason of its status as a publication and/or patent disclosing, or a system on sale and/or in public knowledge or use in the United States embodying the claimed invention before Plaintiffs’ asserted priority date of the ’767 patent. For each item of prior art, Defendants have attached a claim chart identifying where each claim limitation is found in the cited prior art.

Table B: Prior Art to the ’767 Patent

Prior Art to ’767	Anticipation/ Obviousness	Claim Chart
EMC VPLEX System	A & O	Exhibit 767-01
IBM SAN Volume Controller System	A & O	Exhibit 767-02
Microsoft Server System	A & O	Exhibit 767-03
U.S. Patent Application Publication No. 2007/0050538 to Northcutt et al. (“Northcutt”); filed on August 24, 2006; published on March 1, 2007.	A & O	Exhibit 767-04
U.S. Patent No. 8,966,172 to Malwankar (“Malwankar”); filed on January 23, 2012; issued on February 24, 2015.	A & O	Exhibit 767-05
U.S. Patent Application Publication No. 2013/0073747 to Klughart (“Klughart”); filed on September 21, 2011; published on March 21, 2013.	O	Exhibit 767-06
U.S. Patent No. 9,129,068 to Yendigiri et al. (“Yendigiri”); filed on January 4, 2013; issued on September 8, 2015.	O	Exhibit 767-07

Prior Art to ’767	Anticipation/ Obviousness	Claim Chart
U.S. Patent No. 8,321,650 to Steinmetz et al. (“Steinmetz”); filed on November 26, 2007; issued on November 27, 2012.	O	Exhibit 767-08
U.S. Patent Application Publication No. 2015/0222705 to Stephens (“Stephens”); provisional application filed on September 6, 2012; published on August 6, 2015.	O	Exhibit 767-09
U.S. Patent No. 9,424,224 to Voorhees et al. (“Voorhees ’224”); filed on June 27, 2013; issued on August 23, 2016.	O	Exhibit 767-10
U.S. Patent Application Publication No. 2012/0124256 to Johnson et al. (“Johnson”); filed on November 15, 2010; published on May 17, 2012.	O	Exhibit 767-11
U.S. Patent No. 8,539,096 to Voorhees et al. (“Voorhees ’096”); filed on September 26, 2003; issued on September 17, 2013.	O	Exhibit 767-12

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the Asserted Claims of the ’767 patent, including for the reasons illustrated in the accompanying claim chart and based on Plaintiffs’ infringement allegations and contentions for the ’767 patent.

2. Obviousness

A person of ordinary skill in the art would have been motivated to combine any reference in Table B with any other reference(s) in Table B. To the extent that any prior art identified in Table B above is deemed to not anticipate an Asserted Claim, the prior art in Table B may be combined with one or more prior art references in Table B to render the claim obvious. All of these items of prior art are directed to data storage device management and propose solutions similar to those discussed in the ’767 patent.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **Northcutt + Malwankar**
- **Northcutt + Steinmetz**
- **Northcutt + Stephens**

Northcutt discloses a scalable storage system that employs an "advanced storage system" to represent physical storage devices as virtual drives. ¶ [0004]. The advanced storage system acts as an intermediary between the host and the storage devices, connected in a cascaded topology. ¶[0020]. A POSITA would have recognized that enabling simultaneous bidirectional data transfer within such a cascaded storage system topology would be desirable and beneficial, as it improves communication efficiency, load balancing, and fault tolerance. Therefore, a POSITA would naturally have looked to other references that provide details regarding full-duplex communication protocols for storage systems to achieve simultaneous bidirectional data transfer. For example, a POSITA would have been motivated to combine Northcutt with Malwankar, Stephens or Steinmetz, each of which discloses full-duplex communication protocols for storage systems, such as Fibre Channel.

- **Northcutt + Malwankar + Yendigiri**
- **Northcutt + Malwankar + Voorhees '224**
- **Northcutt + Stephens + Yendigiri**
- **Northcutt + Stephens + Voorhees '224**

The combination of Northcutt and Malwankar discloses a data storage system that supports various storage devices, including SATA devices. A POSITA would naturally have looked to other references that provide details for other types of storage devices, particularly

SAS devices, as SAS interfaces support both SAS and SATA devices, allowing seamless integration of SATA storage devices already supported by Northcutt. For example, a POSITA would have been motivated to further combine Northcutt and Malwankar with Yendigiri or Voorhees ’224, each of which is directed to SAS devices.

The combination of Northcutt and Stephens similarly discloses a data storage system that supports various storage devices, including SATA devices. Therefore, for the same reason as above, a POSITA would have been motivated to further combine Northcutt and Stephens with Yendigiri or Voorhees ’224.

- **Northcutt + Steinmetz + Johnson + Voorhees ’096**
- **Northcutt + Malwankar + Yendigiri + Johnson + Voorhees ’096**
- **Northcutt + Malwankar + Voorhees ’224 + Johnson + Voorhees ’096**
- **Northcutt + Stephens + Yendigiri + Johnson + Voorhees ’096**
- **Northcutt + Stephens + Voorhees ’224 + Johnson + Voorhees ’096**

The combination of Northcutt and Steinmetz provides a scalable data storage system that supports SAS devices. A POSITA would naturally have looked to other references that provide details for SAS device management to enhance the reliability and robustness of the system, including details for device discovery in SAS topology. For example, both Voorhees ’096 and Johnson are directed to device discovery in SAS topology. Voorhees ’096 discloses updating routing tables in response to changes detected during a device discovery. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. Therefore, a POSITA would have been motivated to further combine Northcutt and Steinmetz with Voorhees ’096 and Johnson to achieve a robust and reliable storage system. For the same reason, a POSITA would have been motivated to

further combine Northcutt, Malwankar and Yendigiri with Voorhees ’096 and Johnson, further combine Northcutt, Malwankar and Voorhees ’224 with Voorhees ’096 and Johnson, further combine Northcutt, Stephens and Yendigiri with Voorhees ’096 and Johnson, and further combine Northcutt, Stephens and Voorhees ’224 with Voorhees ’096 and Johnson.

- **Malwankar + Klughart**

Malwankar describes its technical field as “data storage” and discloses a processor-agnostic PCIe-based data storage system that routes storage requests directly to storage devices. 1:13-47. It teaches a plurality of storage controllers and assigning storage devices to one or more controllers. 6:54-62. A POSITA would have found it obvious that not all storage devices would be associated with each storage controller, and that in the event that a storage controller receives a host command that directs to an unassociated storage device, it would be desirable to communicate with the controller associated with the target storage device and forward the command. Therefore, a POSITA would naturally have looked to other references that provide details for inter-controller communication. For example, a POSITA would have been motivated to combine Malwankar with Klughart, which has the same technical field as “computer data storage” and discloses methods for communication and command transfer among daisy-chained storage controllers.

- **Malwankar + Yendigiri**
- **Malwankar + Steinmetz**
- **Malwankar + Voorhees ’224**
- **Malwankar + Klughart + Yendigiri**
- **Malwankar + Klughart + Steinmetz**
- **Malwankar + Klughart +Voorhees ’224**

Additionally, Malwankar teaches converting storage-based requests into a format compatible with the target device. 1:58-61. A POSITA would have found it obvious that the protocol conversion disclosed in Malwankar is not limited to PCIe based devices and can be applied to other well-known storage devices, such as SAS devices. Therefore, a POSITA would naturally have looked to other references that provide details for other types of storage devices. For example, a POSITA would have been motivated to combine Malwankar with Voorhees '224, which discloses converting PCIe requests into SAS requests. Similarly, a POSITA would have been motivated to combine Malwankar with Yendigiri or Steinmetz, each of which is directed to SAS devices. For the same reason, a POSITA would have been motivated to further combine Malwankar and Klughart with Yendigiri, Steinmetz or Voorhees '224.

- **Malwankar + Yendigiri + Johnson + Voorhees '096**
- **Malwankar + Steinmetz + Johnson + Voorhees '096**
- **Malwankar + Voorhees '224 + Johnson + Voorhees '096**
- **Malwankar + Klughart + Yendigiri + Johnson + Voorhees '096**
- **Malwankar + Klughart + Steinmetz + Johnson + Voorhees '096**
- **Malwankar + Klughart + Voorhees '224 + Johnson + Voorhees '096**

The combination of Malwankar and Yendigiri provides a scalable data storage system that supports SAS devices. A POSITA would naturally have looked to other references that provide details for SAS device management to enhance the reliability and robustness of the system, including details for device discovery in SAS topology. For example, both Voorhees '096 and Johnson are directed to device discovery in SAS topology. Voorhees '096 discloses updating routing tables in response to changes detected during a device discovery. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and

identifies events that could trigger such changes. Therefore, a POSITA would have been motivated to further combine Malwankar and Yendigiri with Voorhees ’096 and Johnson to achieve a robust and reliable storage system. For the same reason, a POSITA would have been motivated to further combine Malwankar and Steinmetz with Voorhees ’096 and Johnson, further combine Malwankar and Voorhees ’224 with Voorhees ’096 and Johnson, further combine Malwankar, Klughart, and Yendigiri with Voorhees ’096 and Johnson, further combine Malwankar, Klughart, and Steinmetz with Voorhees ’096 and Johnson, and further combine Malwankar, Klughart, and Voorhees ’224 with Voorhees ’096 and Johnson.

- **EMC VPLEX System + Yendigiri**
- **EMC VPLEX System + Steinmetz**
- **EMC VPLEX System + Voorhees ’224**

EMC VPLEX System discloses a scalable storage virtualization system that integrates diverse device storage arrays to enable high availability and seamless data mobility across storage networks. *See* Ex. 767-01. A POSITA would have naturally looked into other references that provide details for various storage devices to complement EMC VPLEX System’s scalable and heterogeneous storage virtualization framework, including SAS devices. For example, a POSITA would have been motivated to combine EMC VPLEX System with Yendigiri, Steinmetz, or Voorhees ’224, either of which is directed to SAS devices.

- **EMC VPLEX System + Yendigiri + Johnson + Voorhees ’096**
- **EMC VPLEX System + Steinmetz + Johnson + Voorhees ’096**
- **EMC VPLEX System + Voorhees ’224 + Johnson + Voorhees ’096**

The combination of EMC VPLEX System and Yendigiri provides a scalable data storage system that supports SAS devices. A POSITA would naturally have looked to other references

that provide details for SAS device management to enhance the reliability and robustness of the system, including details for device discovery in SAS topology. For example, both Voorhees ’096 and Johnson are directed to device discovery in SAS topology. Voorhees ’096 discloses updating routing tables in response to changes detected during a device discovery. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. Therefore, a POSITA would have been motivated to further combine EMC VPLEX System and Yendigiri with Voorhees ’096 and Johnson to achieve a robust and reliable storage system. For the same reason, a POSITA would have been motivated to further combine EMC VPLEX System and Steinmetz with Voorhees ’096 and Johnson, and further combine EMC VPLEX System and Voorhees ’224 with Voorhees ’096 and Johnson.

- **EMC VPLEX System + Yendigiri + Northcutt**
- **EMC VPLEX System + Yendigiri + Klughart**
- **EMC VPLEX System + Steinmetz+ Northcutt**
- **EMC VPLEX System + Steinmetz + Klughart**
- **EMC VPLEX System + Voorhees ’224 + Northcutt**
- **EMC VPLEX System + Voorhees ’224 + Klughart**

The combination of EMC VPLEX System and Yendigiri provides a scalable data storage system that supports SAS devices. In such a system, one EMC VPLEX engine needs to communicate with other EMC VPLEX engines within the cluster or across clusters to coordinate storage operations. A POSITA would have recognized that such communication requires efficient and scalable command execution mechanisms to manage distributed systems effectively. Therefore, a POSITA would have naturally looked into other references that provide

details for resolving a single command into multiple commands to achieve distributed operations across multiple EMC VPLEX engines. For example, a POSITA would have been motivated to further combine EMC VPLEX System and Yendigiri with Northcutt or Klughart, each of which teaches one command transforming into more commands, to enhance EMC VPLEX System's capability of managing distributed tasks efficiently. Similarly, a POSITA would have been motivated to further combine EMC VPLEX System and Steinmetz with Northcutt or Klughart, and further combine EMC VPLEX System and Voorhees '224 with Northcutt or Klughart.

- **IBM SAN Volume Controller System + Klughart**
- **IBM SAN Volume Controller System+ Northcutt**

IBM SAN Volume Controller System discloses a storage virtualization system that manages and optimizes resources across multiple storage arrays, including SAS devices. *See* Ex. 767-02. In such a system, one SAN Volume Controller needs to communicate with other SAN Volume Controllers to provide the host seamless access to virtualized storages. A POSITA would have recognized the need for mechanisms to transform a single command into multiple coordinated operations across nodes to achieve distributed operations and more efficient functionality. For example, a POSITA would have been motivated to combine SAN Volume Controller System with Northcutt or Klughart, each of which teaches one command transforming into more commands, to enhance SAN Volume Controller's capability of managing distributed tasks efficiently.

For any of the system art (or the foregoing combinations that include system art), a POSITA would have found it obvious to implement the prior art's teachings relevant to the claimed master controller(s) or edge controller(s) using ASICs, a well-known technique as shown in Northcutt, Malwankar, and Klughart. A POSITA would have found it obvious to

implement such functionality in the form of an application-specific integrated circuit for many reasons including that an ASIC can provide better performance or be more efficient that an implementation that does not use an ASIC (e.g., software implementation running on a general-purpose processor).

C. U.S. Patent No. 10,873,753

The following table identifies the prior art for the ’753 patent. Each item below constitutes prior art under one or more provisions of 35 U.S.C. §§ 102(a)(1)-(2) by reason of its status as a publication and/or patent disclosing, or a system on sale and/or in public knowledge or use in the United States embodying the claimed invention before Plaintiffs’ asserted priority date of the ’753 patent. For each item of prior art, Defendants have attached a claim chart identifying where each claim limitation is found in the cited prior art.

Table C: Prior Art to the ’753 Patent

Prior Art to ’753	Anticipation/ Obviousness	Claim Chart
Cray Supercomputer System(s)	A & O	Exhibit 753-01
BreakingPoint Scalable System(s)	A & O	Exhibit 753-02
SeaMicro Computing System(s)	A & O	Exhibit 753-03
U.S. Patent Application Publication No. 2013/0346667 to Stroud et al. (“Stroud”); filed on June 21, 2012; published on December 26, 2013	A & O	Exhibit 753-04
U.S. Patent No. 9,306,816 to Jana et al. (“Jana”); filed on December 24, 2013; issued on April 5, 2016	O	Exhibit 753-05
U.S. Patent No. 10,673,783 to Forrest Lawrence Pierson (“Pierson”); filed on January 29, 2019; continuation of application No. 15/202,379 filed on July 5, 2016; parent application filed on September 9, 2013; claims priority to provisional application Nos. 62/188,731, filed on July 5, 2015 and 61/778,393, filed on March 12, 2013.	A & O	Exhibit 753-06

Prior Art to ’753	Anticipation/ Obviousness	Claim Chart
U.S. Patent Number. 9,148,389 to Da Silva et al. (“Da Silva”); filed November 12, 2012; issued on September 29, 2015	O	Exhibit 753-07
Harvey G. Cragon, “Memory Systems and Pipelined Processors”; published at least by 1996	O	Exhibit 753-08
Information Science Institute, “Internet Protocol: DARPA Internet Program Protocol Specification” (“RFC 791”); published at least by September 1981	O	Exhibit 753-09
U.S. Publication No. 2002/0087821 A1 to Saulsbury et al. (“Saulsbury”); published in July 4, 2002	O	Exhibit 753-10
U.S. Patent No. 5,155,833 to Cullison et al. (“Cullison”); issued on October 13, 1992	O	Exhibit 753-11

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the Asserted Claims of the ’753 patent, including for the reasons illustrated in the accompanying claim chart and based on Plaintiffs’ infringement allegations and contentions for the ’753 patent.

2. Obviousness

A person of ordinary skill in the art would have been motivated to combine any reference in Table C with any other reference(s) in Table C. To the extent that any prior art identified in Table C above is deemed to not anticipate an Asserted Claim, the prior art in Table C may be combined with one or more prior art references in Table C to render the claim obvious. All of these items of prior art are directed to and propose solutions similar to those discussed in the ’753 patent, and is in the same field. All of these items of prior art are directed to computer

architecture and/or dataflow processing, and propose solutions similar to those discussed in the ’753 patent.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **Cray Supercomputer System(s) + Pierson or Stroud**
- **Any of the above + Da Silva**
- **Any of the above + Cragon**
- **Any of the above + Saulsbury or Cullison**

The Cray Supercomputer System(s) discloses a supercomputer computer architecture with swappable cards in a chassis-based networked topology for distributed computing. *See, e.g.*, Exhibit 753-01. A POSITA would naturally have looked to other references that provide details for networked computing environments, including details and features for chassis, cards, and networks in data centers. For example, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with Pierson, which teaches a similar multi-card architecture for parallel, distributed processing in a networked environment, with details and features for connecting cards using a backplane and other networks, and regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Pierson, Abstract, 3:35-4:2, 4:3-26, 7:1-32, 14:50-64, 15:2-13, 19:40-53, 20:46-55, 27:44-28:15, FIGs. 1, 2, 5, 8. Likewise, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with Stroud, which teaches a multi-card chassis architecture for network processing and simulation, with details and features regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Stroud, ¶¶ [0098]-[0102], [0322]-[0342] FIGs. 3, 54, 55A, 55B, 56A. Similarly, a POSITA

would have been motivated to combine the Cray Supercomputer System(s) with Da Silva, which teaches a multi-chassis system in a networked environment with details and features (e.g., regarding packet format and headers) for routing packets in a multi-chassis architecture. *See* Da Silva, Abstract, 4:23-33, 4:52-54, 5:13-18, 9:1-18, FIGs. 4 and 6. A POSITA would naturally have looked to these references and been motivated to combine them with the Cray Supercomputer System(s) because, e.g., they provide additional implementation details for chassis architectures in a networked setting; facilitate communication between cards and processors, allowing more granular access to particular components; reduce network overhead in distributed computing systems with multiple cards and chassis; and improve network efficiency in such systems.

Additionally, the Cray Supercomputer System(s) supports an assortment of memory functions and types, including L1, L2, and L3 cache, DDR memory, storage arrays, flash memory, and disk drives. *See, e.g.*, Exhibit 753-01. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with the Cray Supercomputer System(s) because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce memory

latency for applications where latency a the bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

- **BreakingPoint Scalable System(s)+ Pierson or Stroud**
- **Any of the above + Da Silva**
- **Any of the above + Cragon**
- **Any of the above + Saulsbury or Cullison**

The BreakingPoint Scalable System(s) discloses a chassis-based architecture with swappable blades for use in networked environments with various networking protocols for distributed computing, including with streams of packet data. *See, e.g.*, Exhibit 753-02. A POSITA would naturally have looked to other references that provide details for networked computing environments, including details and features for chassis, cards/blades, and networks to support data streams. For example, a POSITA would have been motivated to combine the BreakingPoint Scalable System(s) with Pierson, which teaches a multi-card architecture for parallel, distributed processing in a networked environment, with details and features for connecting cards using a backplane and other networks, and regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Pierson, Abstract, 3:35-4:2, 4:3-26, 7:1-32, 14:50-64, 15:2-13, 19:40-53, 20:46-55, 27:44-28:15, FIGs. 1, 2, 5, 8. Likewise, a POSITA would have been motivated to combine the BreakingPoint Scalable System(s) with Stroud, which teaches a multi-card chassis architecture for network processing and simulation, with details and features regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Stroud, ¶¶ [0098]-[0102], [0322]-[0342] FIGs. 3, 54, 55A, 55B, 56A. Similarly, a POSITA would have

been motivated to combine the BreakingPoint Scalable System(s) with Da Silva, which teaches a scalable multi-chassis system in a networked environment with details and features (e.g., regarding packet format and headers) for routing packets in a multi-chassis architecture. *See* Da Silva, Abstract, 4:23-33, 4:52-54, 5:13-18, 9:1-18, FIGs. 4 and 6. A POSITA would naturally have looked to these references and been motivated to combine them with the BreakingPoint Scalable System(s) because, e.g., they provide additional implementation details for chassis architectures in a networked setting; facilitate communication between cards and processors, allowing more granular access to particular components; reduce network overhead in distributed computing systems with multiple cards and chassis; and improve network efficiency in such systems.

Additionally, the BreakingPoint Scalable System(s) supports an assortment of memory functions and types, including RAM and hard disks. *See, e.g.*, Exhibit 753-02. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine the BreakingPoint Scalable System(s) with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with the BreakingPoint Scalable System(s) because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce memory latency for applications where latency a the

bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

- **SeaMicro Computing System(s)+ Pierson or Stroud**
- **Any of the above + Da Silva**
- **Any of the above + Cragon**
- **Any of the above + Saulsbury or Cullison**

The SeaMicro Computing System(s) discloses an energy-efficient, high-bandwidth microserver architecture with swappable cards in a dense chassis configuration for networked, distributed computing. *See, e.g.*, Exhibit 753-03. A POSITA would naturally have looked to other references that provide details for networked computing environments, including details and features for chassis, cards, and networks to provide "rapid, reliable, and scalable" infrastructure to "application[s]; and enterprises." For example, a POSITA would have been motivated to combine the SeaMicro Computing System(s) with Pierson, which teaches a multi-card architecture for parallel, distributed processing in a networked environment, with details and features for connecting cards using a backplane and other networks, and regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Pierson, Abstract, 3:35-4:2, 4:3-26, 7:1-32, 14:50-64, 15:2-13, 19:40-53, 20:46-55, 27:44-28:15, FIGs. 1, 2, 5, 8. Likewise, a POSITA would have been motivated to combine the SeaMicro Computing System(s) with Stroud, which teaches a multi-card chassis architecture for network processing and simulation, with details and features regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards. *See* Stroud, ¶¶ [0098]-[0102], [0322]-[0342] FIGs. 3, 54, 55A, 55B, 56A. Similarly, a POSITA

would have been motivated to combine the SeaMicro Computing System(s) with Da Silva, which teaches a scalable multi-chassis system in a networked environment with details and features (e.g., regarding packet format and headers) for routing packets in a multi-chassis architecture. *See* Da Silva, Abstract, 4:23-33, 4:52-54, 5:13-18, 9:1-18, FIGs. 4 and 6. A POSITA would naturally have looked to these references and been motivated to combine them with the SeaMicro Computing System(s) because, e.g., they provide additional implementation details for chassis architectures in a networked setting; facilitate communication between cards and processors, allowing more granular access to particular components; reduce network overhead in distributed computing systems with multiple cards and chassis; and improve network efficiency in such systems.

Additionally, the SeaMicro Computing System(s) supports an assortment of memory functions and types, including L1, L2, and L3 cache, DRAM, SATA SSD drives, hard disk drives, DDR3, and SODIMM. *See, e.g.*, Exhibit 753-03. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine the SeaMicro Computing System(s) with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with the SeaMicro Computing System(s) because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce

memory latency for applications where latency a the bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

- **Stroud + Da Silva**
- **Stroud + RFC 791**
- **Any of the above + Cragon**
- **Any of the above + Saulsbury or Cullison**

Stroud teaches a chassis-based architecture with swappable cards for networked environments and distributed computing, including support for various networking protocols and the generation and processing of streams of packet data. *See* Stroud, ¶¶ [0098]-[0102], [0322]-[0342] FIGs. 3, 54, 55A, 55B, 56A. Stroud further teaches details regarding packet headers sent between chassis components. *See, e.g.,* Exhibit 753-04. Therefore, a POSITA looking at Stroud's multi-chassis architecture would naturally have looked to other references that provide details for networked computing environments, including details and features for packet routing in a multi-chassis architecture. For example, a POSITA would have been motivated to combine Stroud with Da Silva, which teaches a multi-chassis architecture with details and features, including packet format and headers, for routing packets in a multi-chassis architecture. *See* Da Silva, Abstract, 4:23-33, 4:52-54, 5:13-18, 9:1-18, FIGs. 4 and 6. A POSITA would naturally have looked combine Da Silva with Stroud because, e.g., the combination would have provided additional implementation details for chassis architectures in a networked setting; facilitated communication between cards and processors, allowed more granular access to particular

components; reduced network overhead in distributed computing systems with multiple cards and chassis; and improved network efficiency in such systems.

Stroud teaches the use of IP packets, which Plaintiffs have accused of infringement. Since Stroud already teaches IP packets, a POSITA would have been motivated to combine Stroud with the standards document for IP packets, RFC 791, which teaches details regarding the packet format and headers. *See* RFC 791 at 13, 17-20, 29-30, 35, 38.

Additionally, Stroud supports an assortment of memory functions and types, including flash memory, HDDR, DDR2, and DDR3. *See* Stroud, ¶¶ [0103], [0114], [0127], [0169], [0717], FIGs. 4-5, 14A-14B, 55A-55B. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine Stroud with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with Stroud because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce memory latency for applications where latency a the bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

- **Jana + RFC 791**
- **Jana + RFC 791 + Cragon**
- **Any of the above + Saulsbury or Cullison**

Jana discloses a chassis-based architecture with swappable cards for networked environments and distributed computing, including support for various networking protocols and the generation and processing of streams of packet data. *See* Jana, 4:51-5:45, 6:8-33, FIGs. 2-3. Jana teaches the use of IP packets, which Plaintiffs have accused of infringement. Since Jana already teaches IP packets, a POSITA would have been motivated to combine Jana with the standards document for IP packets, RFC 791, which teaches details regarding the packet format and headers. *See* RFC 791 at 13, 17-20, 29-30, 35, 38.

Additionally, Jana supports an assortment of memory functions and types, including flash memory and RAM. *See* Jana, 6:33-59, 7:5-26, FIG. 3. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine Jana with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with Jana because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce memory latency for applications where latency a the bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on

performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

- **Pierson + Stroud**
- **Any of the above + Cragon**
- **Any of the above + Saulsbury or Cullison**

Pierson discloses a multi-card architecture for parallel, distributed processing in a networked environment, with details and features for connecting cards using a backplane and other networks. *See* Pierson, Abstract, 3:35-4:2, 4:3-26, 7:1-32, 14:50-64, 15:2-13, 19:40-53, 20:46-55, 27:44-28:15, FIGs. 1, 2, 5, 8. A POSITA looking to implement a multi-card chassis architecture for distributed computing or packet processing would naturally have looked to other references that provide details for chassis-based architectures, cards, and networks in distributed computing. For example, a POSITA would have been motivated to combine Pierson with Stroud, which teaches a multi-card chassis architecture for network processing and testing, with details and features regarding the configuration of chassis systems and the packet format and headers for systems having multiple cards packet format and headers for routing packets in a network environment including a multi-card chassis architecture. *See* Stroud, ¶¶ [0098]-[0102], [0322]-[0342] FIGs. 3, 54, 55A, 55B, 56A. A POSITA would naturally have looked to Stroud and been motivated to combine Stroud with Pierson because, e.g., the combination would have provided additional implementation details for chassis architectures in a networked setting; facilitated communication between cards and processors, allowed more granular access to particular components; reduced network overhead in distributed computing systems with multiple cards and chassis; and improved network efficiency in such systems.

Additionally, Pierson supports an assortment of memory functions and types, including cache, non-volatile memory, and DRAM. *See* Pierson 20:56-21:23, 21:23-27, 21:60-22:3, 22:20-38, FIG. 4. A POSITA would naturally have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine Pierson with Cragon, which teaches memory architectures and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491), and with Saulsbury and/or Cullison, which each teaches the selective operation of memory in different configurations. *See* Saulsbury, Abstract, ¶¶ [0003], [0005]-[0006], [0008], [0017], [0020], [0022], [0024]-[0026], [0068]-[0070], FIGs. 1 and 5; Cullison, 1:11-23, 1:41-63, 2:18-32, 2:36-48, 2:58-62, 2:63-3:21, FIG. 1. A POSITA would naturally have looked to these references and been motivated to combine them with the Pierson because, e.g., the combinations would provide for more flexible memory arrangements to account for different usage models and applications; reduce memory latency for applications where latency a the bottleneck on performance; improve memory throughput for applications where bandwidth is a bottleneck on performance; improve efficiency and utilization of available memory; and increase performance of distributed computing systems.

D. U.S. Patent No. 10,985,943

The following table identifies the prior art for the ’943 patent. Each item below constitutes prior art under one or more provisions of 35 U.S.C. §§ 102(a)(1)-(2) by reason of its status as a publication and/or patent disclosing, or a system on sale and/or in public knowledge or use in the United States embodying the claimed invention before Plaintiffs’ asserted priority date of the ’943 patent. For each item of prior art, Defendants have attached a claim chart identifying where each claim limitation is found in the cited prior art.

Table D: Prior Art to the ’943 Patent

Prior Art to ’943	Anticipation/ Obviousness	Claim Chart
U.S. Patent No. 5,784,577 to Jacobson et al. (“Jacobson”); filed on August 5, 1996; issued on July 21, 1998.	O	Exhibit 943-01
U.S. Patent No. 6,429,682 to Schultz et al. (“Schultz”); filed on May 25, 2001; issued on August 6, 2002.	O	Exhibit 943-02
Pogliani et al., “Quality of Service Driven Runtime Resource Allocation in Reconfigurable HPC Architectures” (“Pogliani”); presented on August 24-26, 2016; published online at least by July 17, 2017.	O	Exhibit 943-03
U.S. Patent Application Publication No. 2018/0027062A1 to Bernat et al. (“Bernat”); filed on June 30, 2017; published on Jan 25, 2018.	O	Exhibit 943-04
U.S. Patent Application Publication No. 2015/0109024A1 to Abdelfattah et al. (“Abdelfattah”); filed on October 22, 2013; published on April 23, 2015.	O	Exhibit 943-05
U.S. Patent No. 8,461,869 to Rahim et al. (“Rahim”); filed on August 19, 2011; issued on June 11, 2013.	O	Exhibit 943-06
Cervero et al., “A resource manager for dynamically reconfigurable FPGA-based embedded systems” (“Cervero”); presented on September 4-6, 2013; published online at least by October 15, 2013.	O	Exhibit 943-07
Cray Supercomputer System(s)	A & O	Exhibit 943-08
AWS EC2 System	A & O	Exhibit 943-09

Prior Art to ’943	Anticipation/ Obviousness	Claim Chart
MS Cloud Platform Solutions System(s)	A & O	Exhibit 943-10
UToronto SAVI System	A & O	Exhibit 943-11
IBM HPC System(s)	A & O	Exhibit 943-12
U.S. Patent No. 10,776,142 to Seshadri et al. (“Seshadri”); filed on December 13, 2017; issued on September 15, 2020.	A & O	*See* IPR2025-00223
U.S. Patent Application Publication No. 2019/0095113A1 to Atsatt et al. (“Atsatt-113”); filed on September 28, 2017; published on Jan 25, 2018.	A & O	
U.S. Patent Application Publication No. 2018/0191642A1 to Biederman et al. (“Biederman”); filed on December 30, 2016; published on July 5, 2018.	O	
Tarafdar et al., “Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center” (“Tarafdar”); presented on February 22-24, 2017; published February 22, 2017.	O	
U.S. Patent No. 8,015,386 to Kulkarni et al. (“Kulkarni”); filed on March 31, 2008; issued on September 6, 2011.	O	
U.S. Patent Application Publication No. 2019/0041923A1 to Atsatt et al. (“Atsatt-923”); filed on December 27, 2017; published on February 7, 2019.	O	

In addition to the anticipation and obviousness explained in this section, the ’943 patent is anticipated and/or rendered obvious by prior art as explained by Defendants’ Petition for Inter Partes Review (assigned Case No. IPR2025-00223), which is incorporated herein by reference:

Grounds	Claims	Prior Art
1	1-5, 10-14	Seshadri
2	1-5, 9-14, 18	Seshadri further in view of Biederman
3	1-5, 10-14	Atsatt-113
4	1-5, 9-14, 18	Atsatt-113 further in view of Biederman
5	6, 15	Grounds 1-4 further in view of Tarafdar
6	7, 16	Grounds 1-4 further in view of Kulkarni
7	8, 17	Grounds 1-4 further in view of Atsatt-923

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the Asserted Claims of the ’943 patent, including for the reasons illustrated in the accompanying claim chart and based on Plaintiffs’ infringement allegations and contentions for the ’943 patent.

2. Obviousness

A person of ordinary skill in the art would have been motivated to combine any reference in Table D with any other reference(s) in Table D. To the extent that any prior art identified in Table D above is deemed to not anticipate an Asserted Claim, the prior art in Table D may be combined with one or more prior art references in Table D to render the claim obvious. All of these items of prior art are directed to PLDs and data processing systems and propose solutions similar to those discussed in the ’943 patent.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **AWS EC2 System + Jacobson, Schultz, or Seshadri**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

For the AWS EC2 System, a POSITA would have naturally looked to other references that provide details related to cloud computing and networked servers, virtual machines, the management of distributed processing resources, and the composition and operation of FPGAs, as explained below.

The AWS EC2 System discloses a PLD, such as an FPGA, for data processing in data-center servers. *See* Ex. 943-09. For example, AWS EC2 Systems provide F1 instances that allow users to configure and run custom logic designs on FPGAs in the AWS Elastic Compute Cloud. *Id.* The F1 instance FPGAs have a Shell (SH) region and a Custom Logic (CL) region. *Id.* The AWS EC2 System restricts access to the CL; to program the CL, users access the CL through API calls, a command line interface (AWS CLI), a web console (AWS Console), or an SDK, among others. Therefore, a POSITA would have been motivated to combine the AWS EC2 System with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, or Seshadri. Jacobson teaches "using a predetermined code access" and "once authorization is confirmed, the PLD is unlocked" for configuration. *See* Ex. 943-01. Schultz teaches a "security control option [that] selectively restricts access to configuration and read operations" and the use of "a frame mask register" "thereby allowing selective access to individual groups of configuration data." *See* Ex. 943-02. Seshadri—which a POSITA would have looked to combine with AWS because it also comes from Amazon—teaches techniques for checking user account permissions before allowing reconfiguration of

client-configurable circuits. *See* IPR2025-00223 (Ground 1). A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security in a networked environment with multiple users loading custom logic designs on shared cloud resources.

As an additional example, the AWS EC2 System provided PLD/FPGA access, for example via F1 instances, in multi-user environments, with users loading their own custom logic. *See* Ex. 943-09. Therefore, a POSITA would have been motivated to combine AWS EC2 with known techniques for utilizing and managing multiple partitions on an FPGA, including support for adjusting the size of partitions, including but not limited to Cervero. Cervero teaches the dynamic allocation, resizing, and combination of multiple partitions within an FPGA. *See* Ex. 943-07. A POSITA would have looked to such references and been motivated to combine because it would have increased the utilization of FPGA resources and reduced costs for the cloud service provider and users by allowing multiple users to load custom logic on the same FPGA, with partitions sized appropriate to the user's logic design.

As an additional example, the AWS EC2 System discloses PLDs, such as FPGAs, for data processing in data-centers. Given the constraints on power, thermal budgets, and network bandwidth in data centers, a POSITA would have looked to combine the AWS EC2 System with known techniques for monitoring/performing analytics of resources and bandwidth utilization on FPGAs, including but not limited to Pogliani or Bernat, which teach such techniques. *See* Ex. 943-03, Ex. 943-04. A POSITA would have looked to and been motivated to combine the AWS EC2 System with such references because it would have improved the efficiency of FPGA resource deployment in distributed computing and cloud settings, while improving and maintaining quality of service.

Additionally, to further address data center constraints on power, thermal budgets, and network bandwidth, a POSITA would have been motivated to combine the AWS EC2 System with known techniques for scaling the voltage and clock frequency of the FPGA, including but not limited to Rahim. Rahim teaches dynamically regulating frequency of discrete portions in an FPGA for thermal and power management. *See* Exhibit Ex. 943-06. A POSITA would have looked to and been motivated to combine the AWS EC2 System with such references because it would have provided higher performance when needed while reducing power consumption and heat generation when higher performance is not needed.

As an additional example, the AWS EC2 System includes FPGAs with various I/O ports for communications. *See* Ex. 943-09. Therefore, a POSITA would have been motivated to combine the AWS EC2 with known techniques for implementing and optimizing communication ports on an FPGA, including but not limited to Abdelfattah. Abdelfattah teaches techniques for using ports with adjustable data width and speed in FPGAs. *See* Ex. 943-05. A POSITA would have looked to and been motivated to combine such references with the AWS EC2 System because it would have provided a flexible means to carry out internal communications, scale bandwidth when needed, and increase the performance of FPGAs.

Additionally, a POSITA would have been motivated to apply teachings from Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 to any of the above-described combinations involving the AWS EC2 System. The Petition in IPR2025-00223 explained the motivation to combine Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations, including for use in data centers. The AWS EC2 System used PLD/FPGA configurations in data centers for cloud computing. *See* Ex. 943-09. Therefore, the motivations to combine explained

in the incorporated Petition (IPR2025-00223) also apply to the AWS EC2 System, and POSITA would have been motivated to combine for similar reasons.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **MS Cloud Platform Solutions System(s) + Jacobson, Schultz, or Atsatt-113**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

For the MSCPS System(s), a POSITA would have naturally looked to other references that provide details related to cloud computing and networked servers, virtual machines, the management of distributed processing resources, and the composition and operation of FPGAs, as explained below.

The MSCPS System(s) discloses a PLD, such as an FPGA, for data processing in data-center servers. *See* Ex. 943-10. For example, MSCPS System(s) provide FPGAs on NICs and accelerator cards in cloud servers. *Id.* The FPGAs have a Shell region and a Role region. *Id.* The MSCPS System(s) restricts access to the Role; to program the Role, users access reconfiguration functions through the shell and must log in to servers hosting the cards. Therefore, a POSITA would have been motivated to combine the MSCPS System(s) with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, or Atsatt-113. Jacobson teaches "using a predetermined code access" and "once authorization is confirmed, the PLD is unlocked" for configuration. *See* Ex. 943-01. Schultz teaches a "security control option [that] selectively restricts access to configuration and read operations" and the use of "a frame mask register" "thereby allowing selective access to

individual groups of configuration data." *See* Ex. 943-02. Atsatt-113 teaches techniques for checking user account permissions before isolating/deisolating a configuration port to allow reconfiguration of user partitions. *See* IPR2025-00223 (Ground 3). Moreover, MSCPS uses Intel/Altera FPGAs in its servers, and Atsatt-113 is a patent from Intel/Altera. A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security in a networked environment with multiple users loading custom logic designs on shared cloud resources.

As an additional example, the MSCPS System(s) provided PLD/FPGA access in multi-user environments with users loading their own custom logic, including for example on the Azure/Catapult SmartNICs. *See* Ex. 943-10. Therefore, a POSITA would have been motivated to combine MSCPS with known techniques for utilizing and managing multiple partitions on an FPGA, including support for adjusting the size of partitions, including but not limited to Cervero. Cervero teaches the dynamic allocation, resizing, and combination of multiple partitions within an FPGA. *See* Ex. 943-07. A POSITA would have looked to such references and been motivated to combine because it would have increased the utilization of FPGA resources and reduced costs for the cloud service provider and users.

As an additional example, the MSCPS System(s) discloses PLDs, such as FPGAs, for data processing in data-centers. Given the constraints on power, thermal budgets, and network bandwidth in data centers, a POSITA would have looked to combine the MSCPS System(s) with known techniques for monitoring/performing analytics of resources and bandwidth utilization on FPGAs, including but not limited to Pogliani or Bernat, which teach such techniques. *See* Ex. 943-03, Ex. 943-04. A POSITA would have looked to and been motivated to combine the MSCPS System(s) with such references because it would have improved the efficiency of FPGA

resource deployment in distributed computing and cloud settings, while improving and maintaining quality of service.

Additionally, to further address data center constraints on power, thermal budgets, and network bandwidth, a POSITA would have been motivated to combine the MSCPS System(s) with known techniques for scaling the voltage and clock frequency of the FPGA, including but not limited to Rahim. Rahim teaches dynamically regulating frequency of discrete portions in an FPGA for thermal and power management. *See* Exhibit Ex. 943-06. Moreover, MSCPS uses Intel/Altera FPGAs in its servers, and Rahim is a patent from Intel/Altera. A POSITA would have looked to and been motivated to combine the MSCPS System(s) with such references because it would have provided higher performance when needed while reducing power consumption and heat generation when higher performance is not needed.

As an additional example, the MSCPS System(s) includes FPGAs with various communication ports between the Shell and Role and between the Shell and external devices. *See* Ex. 943-10. Therefore, a POSITA would have been motivated to combine the MSCPS System(s) with known techniques for implementing and optimizing communication ports on an FPGA, including but not limited to Abdelfattah. Abdelfattah teaches techniques for using ports with adjustable data width and speed in FPGAs. *See* Ex. 943-05. A POSITA would have looked to and been motivated to combine such references with the MSCPS System(s) because it would have provided a flexible means to carry out internal communications, scale bandwidth when needed, and increase the performance of FPGAs.

Additionally, a POSITA would have been motivated to apply teachings from Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 to any of the above-described combinations involving the MSCPS System(s). The Petition in IPR2025-00223 explained the motivation to combine

Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations, including for use in data centers. The MSCPS System(s) used PLD/FPGA configurations in data centers for cloud computing. *See* Ex. 943-10. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the MSCPS System(s), and POSITA would have been motivated to combine for similar reasons.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **SAVI System + Jacobson, Schultz, or Seshadri**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

For the SAVI System, a POSITA would have naturally looked to other references that provide details related to cloud computing and networked servers, virtual machines, the management of distributed processing resources, and the composition and operation of FPGAs, as explained below.

The SAVI System discloses a PLD, such as an FPGA, for data processing in data-center servers. *See* Ex. 943-11. For example, SAVI System provide FPGAs as accelerators as edge node resources in a cloud. *Id.* The FPGAs have an Abstraction Shell region and a FPGA Kernel region. *Id.* The SAVI System restricts access to the FPGA Kernel; to program user logic to a Kernel, users access the Kernel through the virtual machine using, for example, Openstack APIs. Therefore, a POSITA would have been motivated to combine the SAVI System with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, or Seshadri. Jacobson teaches "using a predetermined code access" and

"once authorization is confirmed, the PLD is unlocked" for configuration. *See* Ex. 943-01. Schultz teaches a "security control option [that] selectively restricts access to configuration and read operations" and the use of "a frame mask register" "thereby allowing selective access to individual groups of configuration data." *See* Ex. 943-02. Seshadri teaches techniques for checking user account permissions before allowing reconfiguration of client-configurable circuits. *See* IPR2025-00223 (Ground 1). A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security in a networked environment with multiple users loading custom logic designs on shared cloud resources.

As an additional example, the SAVI System discloses multiple users sharing space on a PLD, such as an FPGA, including for example on the NetFPGA cards. *See* Ex. 943-11. The SAVI System allows user logics to be loaded to partial reconfiguration areas on the same FPGA in the cloud. *Id.* Therefore, a POSITA would have been motivated to combine SAVI with known techniques for managing and adjusting multiple partitions on an FPGA, including but not limited to Cervero. Cervero teaches the dynamic allocation, resizing, and combination of multiple partitions within an FPGA. *See* Ex. 943-07. A POSITA would have looked to such references and been motivated to combine because it would have increased the utilization of FPGA resources and reduced costs for the cloud service provider and users.

As an additional example, the SAVI System discloses PLDs, such as FPGAs, for data processing in data-centers such as the SAVI Testbed. Given the constraints on power, thermal budgets, and network bandwidth in data centers, a POSITA would have looked to combine the SAVI System with known techniques for monitoring/performing analytics of resources and bandwidth utilization on FPGAs, including but not limited to Pogliani or Bernat, which teach

such techniques. *See* Ex. 943-03, Ex. 943-04. A POSITA would have looked to and been motivated to combine the SAVI System with such references because it would have improved the efficiency of FPGA resource deployment in distributed computing and cloud settings, while improving and maintaining quality of service.

Additionally, to further address data center constraints on power, thermal budgets, and network bandwidth, a POSITA would have been motivated to combine the SAVI System with known techniques for scaling the voltage and clock frequency of the FPGA, including but not limited to Rahim. Rahim teaches dynamically regulating frequency of discrete portions in an FPGA for thermal and power management. *See* Exhibit Ex. 943-06. A POSITA would have looked to and been motivated to combine the SAVI System with such references because it would have provided higher performance when needed while reducing power consumption and heat generation when higher performance is not needed.

As an additional example, the SAVI System includes FPGAs with various I/O ports including PCIe and Ethernet. *See* Ex. 943-11. Therefore, a POSITA would have been motivated to combine the SAVI System with known techniques for implementing and optimizing communication ports on an FPGA, including but not limited to Abdelfattah. Abdelfattah teaches techniques for using ports with adjustable data width and speed in FPGAs. *See* Ex. 943-05. A POSITA would have looked to and been motivated to combine such references with the SAVI System because it would have provided a flexible means to carry out internal communications, scale bandwidth when needed, and increase the performance of FPGAs.

Additionally, a POSITA would have been motivated to apply teachings from Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 to any of the above-described combinations involving the SAVI System. The Petition in IPR2025-00223 explained the motivation to combine Biederman,

Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations, including for use in data centers. The SAVI System used PLD/FPGA configurations in data centers for cloud computing. *See* Ex. 943-09. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the SAVI System, and a POSITA would have been motivated to combine for similar reasons.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **IBM HPC System(s) + Jacobson, Schultz, Seshadri, or Atsatt-113**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

For the IBM HPC System(s), a POSITA would have naturally looked to other references that provide details related to high performance computing and networked servers, virtual machines, the management of distributed processing resources, and the composition and operation of FPGAs, as explained below.

The IBM HPC System(s) discloses a PLD, such as an FPGA, for distributing applications to network-attached FPGAs in data centers. *See* Ex. 943-12. For example, IBM HPC System(s) provide FPGAs in cloudFPGA and Power8+CAPI servers. *Id.* The FPGAs have a Shell region and a Role region. *Id.* The IBM HPC System(s) restricts access to the Role; to program the Role, users access reconfiguration functions through the FPGA Manager Core in the Shell. Therefore, a POSITA would have been motivated to combine the IBM HPC System(s) with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, Seshadri, or Atsatt-113. Jacobson teaches "using a predetermined code

access" and "once authorization is confirmed, the PLD is unlocked" for configuration. *See* Ex. 943-01. Schultz teaches a "security control option [that] selectively restricts access to configuration and read operations" and the use of "a frame mask register" "thereby allowing selective access to individual groups of configuration data." *See* Ex. 943-02. Seshadri teaches techniques for checking user account permissions before allowing reconfiguration of client-configurable circuits. *See* IPR2025-00223 (Ground 1). Atsatt-113 teaches techniques for checking user account permissions before isolating/deisolating a configuration port to allow reconfiguration of user partitions. *See* IPR2025-00223 (Ground 3). Moreover, IBM HPC provides Intel/Altera FPGAs as an option in its servers, and Atsatt-113 is a patent from Intel/Altera. A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security in a networked environment with multiple users loading custom logic designs on shared cloud resources.

As an additional example, the IBM HPC System(s) discloses virtualizing the FPGA to allow multiple users to share space for Roles. *See* Ex. 943-12. The IBM HPC System(s) allows custom user logics to be loaded to multiple roles on the same FPGA. *Id.* It provided PLD/FPGA access in multi-user environments, with users loading their own custom logic. *Id.* Therefore, a POSITA would have been motivated to combine IBM HPC with known techniques for utilizing and managing multiple partitions on an FPGA, including support for adjusting the size of partitions, including but not limited to Cervero. Cervero teaches the dynamic allocation, resizing, and combination of multiple partitions within an FPGA. *See* Ex. 943-07. A POSITA would have looked to such references and been motivated to combine because it would have increased the utilization of FPGA resources and reduced costs for the cloud service provider and users.

As an additional example, the IBM HPC System(s) discloses PLDs, such as FPGAs, for data processing in data-centers. Given the constraints on power, thermal budgets, and network bandwidth in data centers, a POSITA would have looked to combine the IBM HPC System(s) with known techniques for monitoring/performing analytics of resources and bandwidth utilization on FPGAs, including but not limited to Pogliani or Bernat, which teach such techniques. *See* Ex. 943-03, Ex. 943-04. A POSITA would have looked to and been motivated to combine the IBM HPC System(s) with such references because it would have improved the efficiency of FPGA resource deployment in distributed computing and cloud settings, while improving and maintaining quality of service.

Additionally, to further address data center constraints on power, thermal budgets, and network bandwidth, a POSITA would have been motivated to combine the IBM HPC System(s) with known techniques for scaling the voltage and clock frequency of the FPGA, including but not limited to Rahim. Rahim teaches dynamically regulating frequency of discrete portions in an FPGA for thermal and power management. *See* Exhibit Ex. 943-06. Moreover, IBM HPC provides Intel/Altera FPGAs as an option its servers, and Rahim is a patent from Intel/Altera. A POSITA would have looked to and been motivated to combine the IBM HPC System(s) with such references because it would have provided higher performance when needed while reducing power consumption and heat generation when higher performance is not needed.

As an additional example, the IBM HPC System(s) includes FPGAs with various ports for communication within the FPGA and between FPGA Nodes. *See* Ex. 943-12. Therefore, a POSITA would have been motivated to combine the IBM HPC System(s) with known techniques for implementing and optimizing communication ports on an FPGA, including but not limited to Abdelfattah. Abdelfattah teaches techniques for using ports with adjustable data

width and speed in FPGAs. *See* Ex. 943-05. A POSITA would have looked to and been motivated to combine such references with the IBM HPC System(s) because it would have provided a flexible means to carry out internal communications, scale bandwidth when needed, and increase the performance of FPGAs.

Additionally, a POSITA would have been motivated to apply teachings from Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 to any of the above-described combinations involving the IBM HPC System(s). The Petition in IPR2025-00223 explained the motivation to combine Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations, including for use in data centers. The IBM HPC System(s) used PLD/FPGA configurations in data centers for cloud computing. *See* Ex. 943-12. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the IBM HPC System(s), and POSITA would have been motivated to combine for similar reasons.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **Cray Supercomputer System(s) + Jacobson, Schultz, or Seshadri**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

For the Cray Supercomputer System(s), a POSITA would have naturally looked to other references that provide details related to cloud computing and networked servers, virtual machines, the management of distributed processing resources, and the composition and operation of FPGAs, as explained below.

The Cray Supercomputer System(s) discloses a PLD, such as an FPGA, for data processing in high-performance computing systems. *See* Ex. 943-08. For example, Cray Supercomputer System(s) provide FPGAs as "user-programmable accelerators" on blade. *Id.* The FPGAs have a control region encompassing a communication core, a HyperTransport interface, and QDR memory interfaces, and a user configurable region. *Id.* Therefore, given the user of FPGAs in a multiuser environment, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, or Seshadri. Jacobson teaches "using a predetermined code access" and "once authorization is confirmed, the PLD is unlocked" for configuration. *See* Ex. 943-01. Schultz teaches a "security control option [that] selectively restricts access to configuration and read operations" and the use of "a frame mask register" "thereby allowing selective access to individual groups of configuration data." *See* Ex. 943-02. Seshadri teaches techniques for checking user account permissions before allowing reconfiguration of client-configurable circuits. *See* IPR2025-00223 (Ground 1). A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security in a networked environment with multiple users loading custom logic designs on shared cloud resources. Indeed, there is teachings, suggestion and motivation in the art for these combinations. For example, the Cray Supercomputer System(s) provide FPGAs on NICs and accelerator cards in cloud servers, where users access reconfiguration functions through the shell and must log in to servers hosting the cards.

As an additional example, the Cray Supercomputer System(s) discloses custom logics being loaded in parallel on a PLD, such as an FPGA cards. *See* Ex. 943-08. The Cray Supercomputer System(s) allows for application acceleration by supporting multiple instances on

the same FPGA. *Id.* It also provided PLD/FPGA access in multi-user environments, with users loading their own custom logic. *Id.* Therefore, a POSITA would have been motivated to combine Cray Supercomputer System(s) with known techniques for managing and adjusting multiple partitions on an FPGA, including but not limited to Cervero. Cervero teaches the dynamic allocation, resizing, and combination of multiple partitions within an FPGA. *See* Ex. 943-07. A POSITA would have looked to such references and been motivated to combine because it would have increased the utilization of FPGA resources and reduced costs for the cloud service provider and users.

As an additional example, the Cray Supercomputer System(s) discloses PLDs, such as FPGAs, for data processing in data centers using, for example, its CrayXC or CrayXD systems. Given the constraints on power, thermal budgets, and network bandwidth in data centers, a POSITA would have looked to combine the Cray Supercomputer System(s) with known techniques for monitoring/performing analytics of resources and bandwidth utilization on FPGAs, including but not limited to Pogliani or Bernat, which teach such techniques. *See* Ex. 943-03, Ex. 943-04. A POSITA would have looked to and been motivated to combine the Cray Supercomputer System(s) with such references because it would have improved the efficiency of FPGA resource deployment in distributed computing and cloud settings, while improving and maintaining quality of service.

Additionally, to further address data center constraints on power, thermal budgets, and network bandwidth, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with known techniques for scaling the voltage and clock frequency of the FPGA, including but not limited to Rahim. Rahim teaches dynamically regulating frequency of discrete portions in an FPGA for thermal and power management. *See* Exhibit Ex. 943-06. A POSITA

would have looked to and been motivated to combine the Cray Supercomputer System(s) with such references because it would have provided higher performance when needed while reducing power consumption and heat generation when higher performance is not needed.

As an additional example, the Cray Supercomputer System(s) includes FPGAs with various I/O ports including PCIe and Ethernet. *See* Ex. 943-08. Therefore, a POSITA would have been motivated to combine the Cray Supercomputer System(s) with known techniques for implementing and optimizing communication ports on an FPGA, including but not limited to Abdelfattah. Abdelfattah teaches techniques for using ports with adjustable data width and speed in FPGAs. *See* Ex. 943-05. A POSITA would have looked to and been motivated to combine such references with the Cray Supercomputer System(s) because it would have provided a flexible means to carry out internal communications, scale bandwidth when needed, and increase the performance of FPGAs.

Additionally, a POSITA would have been motivated to apply teachings from Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 to any of the above-described combinations involving the Cray Supercomputer System(s). The Petition in IPR2025-00223 explained the motivation to combine Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations, including for use in data centers. The Cray Supercomputer System(s) used PLD/FPGA configurations in data centers for cloud computing. *See* Ex. 943-08. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the Cray Supercomputer System(s), and POSITA would have been motivated to combine for similar reasons.

Defendants further incorporate by reference the invalidity grounds described in Defendants' Petition for Inter Partes Review (assigned Case No. IPR2025-00223).

E. U.S. Patent No. 10,944,634

The following table identifies the prior art for the ’634 patent. Each item below constitutes prior art under one or more provisions of 35 U.S.C. §§ 102(a)(1)-(2) by reason of its status as a publication and/or patent disclosing, or a system on sale and/or in public knowledge or use in the United States embodying the claimed invention before Plaintiffs’ asserted priority date of the ’634 patent. For each item of prior art, Defendants have attached a claim chart identifying where each claim limitation is found in the cited prior art.

Table E Prior Art to ’634 Patent

Prior Art to ’634	Anticipation/ Obviousness	Claim Chart
Verizon Communications Inc.’s networking devices (“Verizon Networking Devices”)	A & O	Exhibit 634-01
Verizon Communications Inc.’s networking solutions (“Verizon Networking Solutions”)	A & O	Exhibit 634-02
Verizon Communications Inc.’s Voice over Internet Protocol devices (“Verizon VoIP Products”)	A & O	Exhibit 634-03
Google LLC’s networking products (“Google Networking Products”)	A & O	Exhibit 634-04
U.S. Patent No. 8,578,008 (“Sella”); filed on July 16, 2013; issued on November 5, 2013.	A & O	Exhibit 634-05
U.S. Patent No. 9,032,060 (“Blaisdell”); filed on September 14, 2011; issued on May 12, 2015.	A & O	Exhibit 634-06
U.S. Patent No. 7,415,038 (“Ullmann”); filed on March 29, 2001; issued on August 19, 2008.	O	Exhibit 634-07
U.S. Patent No. 7,742,499 (“Erskine”); filed on August 18, 2005; issued on June 22, 2010.	A & O	Exhibit 634-08
U.S. Patent No. 8,000,276 (“Scherzer”); filed on May 3, 2007; issued on August 16, 2011.	A & O	Exhibit 634-09
U.S. Patent No. 9,661,553 (“Gillett”); filed on July 14, 2015; issued on May 23, 2017.	O	Exhibit 634-10

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the Asserted Claims of the ’634 patent, including for the reasons illustrated in the accompanying claim chart and based on Plaintiffs’ infringement allegations and contentions for the ’634 patent.

2. Obviousness

A person of ordinary skill in the art would have been motivated to combine any reference in Table E with any other reference(s) in Table E. To the extent that any prior art identified in Table E above is deemed to not anticipate an Asserted Claim, the prior art in Table E may be combined with one or more other prior art references in Table E to render the claim obvious. All of these items of prior art are directed to and propose solutions similar to those discussed in the ’634 patent, and is in the same field. All of these items of prior art are directed to networking systems and architectures, and propose solutions similar to those discussed in the ’634 patent.

The following are exemplary combinations of prior art that would have been obvious to a POSITA:

- **Sella + Ullmann**
- **Sella + Erskine**

In addition to the above-mentioned motivations to combine, Sella is “related to service allocation in a data communication network” and discloses “a system [that] adjusts capacity on a network.” *See* Sella, 1:7-8, 1:37-61. Sella discloses “dynamically changing bandwidth allocation.” *See* Sella, 2:60-3:6. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Sella with Ullmann, which teaches monitoring and managing network bandwidth. *See, e.g.*, Ullmann, 4:18-43.

As an additional example, a POSITA would have been motivated to combine Sella with Erskine, which teaches "adaptive bandwidth management for a packet-based communications network." *See, e.g.,* Erskine, 1:34-42.

- **Blaisdell + Scherzer**
- **Blaisdell + Gillett**

In addition to the above-mentioned motivations to combine, Blaisdell "relates to computing devices and network selection. More specifically, it relates to network selection using bandwidth availability." *See* Blaisdell, 1:16-18. Blaisdell discloses receiving location data of devices and enabling network connections based on bandwidth data. *See, e.g.,* Blaisdell, 2:37-47. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Blaisdell with Scherzer, which discloses obtaining location data of networking devices and establishing network connections based on bandwidth data. *See, e.g.,* Scherzer, 2:42-3:43.

As an additional example, a POSITA would have been motivated to combine Blaisdell with Gillett, which discloses establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality. *See, e.g.,* Gillett, Abstract, 9:57-10:5.

- **Erskine + Sella**
- **Erskine + Ullmann**

In addition to the above-mentioned motivations to combine, Erskine "relates generally to communication systems, and more particularly to an apparatus and method for adaptive

bandwidth management for Voice-Over-IP (VOIP) networks." *See* Erskine, 1:6-9. Erskine discloses "a method of adaptive bandwidth management for a packet-based communications network." *See, e.g.,* Erskine, 1:34-42. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Erskine with Sella, which discloses "adjust[ing] capacity on a network and "dynamically changing bandwidth allocation." *See, e.g.*, Sella, 1:7-8, 1:37-38, 2:60-3:6.

As an additional example, a POSITA would have been motivated to combine Erskine with Ullmann, which teaches monitoring and managing network bandwidth. *See, e.g.*, Ullmann, 4:18-43.

- **Scherzer + Blaisdell**
- **Scherzer + Gillett**

Additionally, one of ordinary skill in the art would have been motivated to combine Scherzer with any other reference(s) in Table E. In addition to the above-mentioned motivations to combine, Scherzer "relates to wireless services and wireless access points." *See* Scherzer, 1:14-15. Scherzer discloses obtaining location data of networking devices and establishing network connections based on bandwidth data. *See, e.g.,* Scherzer, 2:42-3:43. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Scherzer with Blaisdell, which discloses receiving location data of devices and enabling network connections based on bandwidth data. *See, e.g.,* Blaisdell, 2:37-47.

As an additional example, a POSITA would have been motivated to combine Scherzer with Gillett, which discloses establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality. *See, e.g.,* Gillett, Abstract, 9:57-10:5.

- **Verizon Networking Devices + Sella**
- **Verizon Networking Devices + Sella + Ullmann**
- **Verizon Networking Devices + Sella + Erskine**
- **Verizon Networking Devices + Blaisdell**
- **Verizon Networking Devices + Blaisdell + Scherzer**
- **Verizon Networking Devices + Blaisdell + Gillett**
- **Verizon Networking Devices + Erskine**
- **Verizon Networking Devices + Erskine + Ullmann**
- **Verizon Networking Devices + Scherzer**
- **Verizon Networking Devices + Scherzer + Gillett**

Verizon Networking Devices are described in numerous documents including those listed in Exhibit 634-01. One of ordinary skill in the art would have understood that these documents describe Verizon Communications Inc.'s networking devices and would have been motivated to combine the listed documents because they all describe Verizon's networking devices, have the same author, are from the same field, related to the same and/or interoperable methods and systems, and/or attempt to solve the same problem.

In addition to the above-mentioned motivations to combine, Verizon Networking Devices disclose methods, devices, and systems for establishing and managing networking connections

using various network parameters to optimize network performance. For example, Verizon Networking Devices allow users "to connect all [their] devices for fast file transfers and smooth media streaming" and provide "Quality of Service support for traffic prioritization." *See, e.g.*, Exhibit 634-01, Verizon 5 or 8 Port Gigabit Desktop Switch. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Verizon Networking Devices with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Ullmann, which teaches monitoring and managing network bandwidth.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Erskine, which teaches adaptive bandwidth management for a packet-based communications network.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Blaisdell, which teaches receiving location data of devices and enabling network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Scherzer, which teaches obtaining location data of networking devices and establishing network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Gillett, which teaches establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality.

- **Verizon Networking Solutions + Sella**
- **Verizon Networking Solutions + Sella + Ullmann**
- **Verizon Networking Solutions + Sella + Erskine**
- **Verizon Networking Solutions + Blaisdell**
- **Verizon Networking Solutions + Blaisdell + Scherzer**
- **Verizon Networking Solutions + Blaisdell + Gillett**
- **Verizon Networking Solutions + Erskine**
- **Verizon Networking Solutions + Erskine + Ullmann**
- **Verizon Networking Solutions + Scherzer**
- **Verizon Networking Solutions + Scherzer + Gillett**

Verizon Networking Solutions are described in numerous documents including those listed in Exhibit 634-02. One of ordinary skill in the art would have understood that these documents describe Verizon Communications Inc.'s networking solutions and would have been motivated to combine the listed documents because they all describe Verizon's networking solutions, have the same author, are from the same field, related to the same and/or interoperable methods and systems, and/or attempt to solve the same problem.

In addition to the above-mentioned motivations to combine, Verizon Networking Solutions disclose devices, methods, systems, and architectures for establishing and managing networking connections using various network parameters to optimize network performance. For example, Verizon Networking Solutions provide "[a]pplication assessments to determine flow, performance thresholds and routing rules[,] [f]easibility assessments and recommendations to improve network performance and control costs[, and] [u]nbiased analyses of all options, including a hybrid network, adding additional bandwidth to the private network or other alternatives." *See, e.g.*, Exhibit 634-02, Build a Wiser WAN - Managed SD WAN with

Application Aware Routing. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Verizon Networking Solutions with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Ullmann, which teaches monitoring and managing network bandwidth.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Erskine, which teaches adaptive bandwidth management for a packet-based communications network.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Blaisdell, which teaches receiving location data of devices and enabling network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Scherzer, which teaches obtaining location data of networking devices and establishing network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Gillett, which teaches establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality.

- **Verizon VoIP Products + Sella**
- **Verizon VoIP Products + Sella + Ullmann**
- **Verizon VoIP Products + Sella + Erskine**
- **Verizon VoIP Products + Blaisdell**

- **Verizon VoIP Products + Blaisdell + Scherzer**
- **Verizon VoIP Products + Blaisdell + Gillett**
- **Verizon VoIP Products + Erskine**
- **Verizon VoIP Products + Erskine + Ullmann**
- **Verizon VoIP Products + Scherzer**
- **Verizon VoIP Products + Scherzer + Gillett**

Verizon VoIP Products are described in numerous documents including those listed in Exhibit 634-03. One of ordinary skill in the art would have understood that these documents describe Verizon Communications Inc.'s VoIP products and would have been motivated to combine the listed documents because they all describe Verizon's VoIP products, have the same author, are from the same field, related to the same and/or interoperable methods and systems, and/or attempt to solve the same problem.

In addition to the above-mentioned motivations to combine, Verizon VoIP Products disclose VoIP related devices, systems, and architectures for establishing and managing connections using various network parameters to optimize network performance. For example, Verizon VoIP Products facilitate end-to-end VoIP connections and provide "excellent voice quality and optimized packet voice streaming over IP networks, enabling voice, fax, and data traffic to be sent over the same IP network." *See, e.g.*, Exhibit 634-03, Verizon T.38 FAX Configuration Guide for AudioCodes MP-11x. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Verizon VoIP Products with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Ullmann, which teaches monitoring and managing network bandwidth.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Erskine, which teaches adaptive bandwidth management for a packet-based communications network.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Blaisdell, which teaches receiving location data of devices and enabling network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Scherzer, which teaches obtaining location data of networking devices and establishing network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Gillett, which teaches establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality.

- **Google Networking Products + Sella**
- **Google Networking Products + Sella + Ullmann**
- **Google Networking Products + Sella + Erskine**
- **Google Networking Products + Blaisdell**
- **Google Networking Products + Blaisdell + Scherzer**
- **Google Networking Products + Blaisdell + Gillett**
- **Google Networking Products + Erskine**
- **Google Networking Products + Erskine + Ullmann**
- **Google Networking Products + Scherzer**
- **Google Networking Products + Scherzer + Gillett**

Google Networking Products are described in numerous documents including those listed in Exhibit 634-04. One of ordinary skill in the art would have understood that these documents describe Google LLC's networking products and would have been motivated to combine the listed documents because they all describe Google's networking products, have the same author, are from the same field, related to the same and/or interoperable methods and systems, and/or attempt to solve the same problem.

In addition to the above-mentioned motivations to combine, Google Networking Products disclose methods, devices, systems, and architectures for establishing and managing connections using various network parameters to optimize network performance. For example, Google Networking Products provide "wired bridge mode and [r]outer functionality" and support features such as "[m]ulti-user sharing a high-speed Internet connection" and "bandwidth control." *See, e.g.*, Exhibit 634-04, Google OnHub User Guide. A POSITA would naturally have looked to other references that provide details related to bandwidth allocation and management, Quality of Service (QoS), load balancing, congestion control, adaptive routing, and/or network analytics and monitoring.

For example, a POSITA would have been motivated to combine Google Networking Products with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Ullmann, which teaches monitoring and managing network bandwidth.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Erskine, which teaches adaptive bandwidth management for a packet-based communications network.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Blaisdell, which teaches receiving location data of devices and enabling network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Scherzer, which teaches obtaining location data of networking devices and establishing network connections based on bandwidth data.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Gillett, which teaches establishing end-to-end network connections, determining locations of networking devices, and measuring connection quality.

II. Identification of Limitations that Defendants Contend Are Indefinite or Lack Written Description 35 U.S.C. § 112

A. U.S. Patent No. 8,924,596

1. "input queue implemented in hardware" / "output queue implemented in hardware"

Under Plaintiffs' Infringement Contentions and apparent application of their interpretation of the claim language, claims 1, 7, 13, and their dependents of the '596 patent are invalid under 35 U.S.C. §112(b) because the claim language, when read in light of the intrinsic record, fails to inform a person of ordinary skill in the art the scope of the claimed invention with reasonable certainty with respect to "input queue implemented in hardware" and "output queue implemented in hardware" claim limitations. Under Plaintiffs' apparent application of these two claim terms to their infringement theories, a person of ordinary skill in the art would not be reasonably certain as the scope of the invention, because Plaintiffs' infringement theories appear to blur hardware and software implementations of queues by encompassing software within the scope of an "input queue implemented in hardware" and "output queue implemented in hardware." Plaintiffs' infringement theories effectively read out of the claim limitation the

words "implemented in hardware" and fails to provide any bounds on the scope of the term "implemented in hardware," rendering both terms indefinite under 35 U.S.C. §112.

B. U.S. Patent No. 9,529,767

1. "abstraction protocol"

Claims 1 and 16, and all claims depending from them, are invalid under 35 U.S.C. §112 because the patent fails to provide a sufficient written description of the claimed "abstraction protocol" and fails to clearly define the term, rendering it indefinite. Claims 1 and 16 recite an "abstraction protocol" that comprises "a full-duplex protocol supporting full command queuing for [] at least one storage media device." However, the specification fails to provide sufficient detail about how the "abstraction protocol" is implemented or how it can achieve these functionalities. While the specification identifies features that the abstraction protocol may possess, such as being "performance-centric" and "link-agnostic," ('767 pat., 4:35-40), and "support[ing] common high-level read and write access to a pool of storage media devices" (*id.*, 4:35-38), it fails to provide sufficient details about the structure, operation or implementation of the "abstraction protocol." For example, the specification does not include concrete examples or diagrams illustrating how to implement the "abstraction protocol." Instead, it merely states that "[t]his invention simplifies a host that accesses SATA and SAS storage media devices by abstracting the SATA and SAS protocols with one full duplex protocol," (*id.*, 3:65-67), which fails to inform with reasonable certainty to one of ordinary skill in the art how the SATA and SAS protocols are abstracted. This lack of detailed explanation further results in ambiguity. Indeed, the term "abstraction protocol" is not precisely defined and is used vaguely throughout the patent without clear boundaries, making the scope of the claims ambiguous. The claims fail to inform a person of ordinary skill in the art about the exact nature of the abstraction protocol or how it distinguishes from existing protocols, leaving them to speculate the scope of the claimed

invention. As a result, the claims are invalid for lack of written description and indefiniteness under 35 U.S.C. §112.

2. "full command queuing"

Claims 1 and 16, and all claims depending from them, are invalid under 35 U.S.C. §112 because the patent fails to provide a sufficient written description of the claimed "full command queueing" and fails to clearly define the term, rendering it indefinite. The claims recite "full command queueing" as a feature of the abstraction protocol, however, the specification fails to provide any explanation of what "full command queuing" entails, how it is implemented, or how it differs from the existing queueing methods such as native command queuing and tagged command queueing. The specification merely mentions "full command queuing" in a conclusory manner, without offering any guidance as to its scope or operation. There is no description of how commands are queued, managed, or executed under this full command queueing context. This lack of detailed explanation further results in ambiguity, as the term "full command queuing" could be subject to speculative interpretation, making it impossible for a person of ordinary skill in the art to understand the full scope of the invention or determine the boundaries of the claimed invention. As a result, the claims are invalid for lack of written description and indefiniteness under 35 U.S.C. §112.

3. "single high level command"

Claims 7 and 19 are invalid under 35 U.S.C. §112 because the patent fails to provide a sufficient written description of the claimed "single high level command" and fails to clearly define the term, rendering it indefinite. The specification mentions the concept of a "single high level command" in the context of the abstraction protocol, however, it fails to provide a detailed explanation of what constitutes a "single high level command" or how it is structured. This lack of detailed explanation further results in ambiguity, as the term "single high level command"

could be subject to speculative interpretation, making it impossible for a person of ordinary skill in the art to understand the full scope of the invention or determine the boundaries of the claimed invention. As a result, the claims are invalid for lack of written description and indefiniteness under 35 U.S.C. §112.

4. "the edge processing element associated with the edge controller authors abstract protocol messages intended for the other edge controller"

Claim 1 and all claims depending from it are invalid under 35 U.S.C. §112 because the patent fails to provide a sufficient written description of the claimed "the edge processing element associated with the edge controller authors abstract protocol messages intended for the other edge controller" and fails to clearly define the term, rendering it indefinite. The specification mentions that the edge processing element "authors" abstract protocol messages, but it fails to explain or define what authoring entails or how it is implemented. This lack of detailed explanation further results in ambiguity, as "authors" could be subject to speculative interpretation, making it impossible for a person of ordinary skill in the art to understand the full scope of the invention or determine the boundaries of the claimed invention. In addition, the specification does not explain or define what is meant by "*intended* for the other edge controller." This lack of detailed explanation further results in ambiguity, as "intended" could be subject to speculative interpretation, making it impossible for a person of ordinary skill in the art to understand the full scope of the invention or determine the boundaries of the claimed invention. As a result, the claims are invalid for lack of written description and indefiniteness under 35 U.S.C. §112.

5. "the master controller" and "the edge controller"

Claims 8, 9, 11, 13, 14, 15 and 19 are invalid under 35 U.S.C. §112 because the patent fails to clearly define the terms "the master controller" and "the edge controller" as recited in the

claims, rendering them indefinite. Claim 1 references "a master controller," "one edge controller," "one other master controller," and "one other edge controller." Therefore, it is unclear which specific master controller and edge controller are being referred to when claims 8, 9, 11, 13, 14, 15 and 19 recite "the master controller" or "the edge controller." Such ambiguity makes it impossible for a person of ordinary skill in the art to understand the full scope of the invention or determine the boundaries of the claimed invention. As a result, the claims are invalid for indefiniteness under 35 U.S.C. §112.

C. U.S. Patent No. 10,873,753

1. "a sandbox"

Claims 3, 10, and 14 are invalid under 35 U.S.C. §112(b). Under Plaintiffs' Infringement Contentions and apparent application of their interpretation of the claim language, these claims are invalid under 35 U.S.C. §112(b) because the claim language, when read in light of the intrinsic record, fails to inform a POSITA of the scope of the claimed invention with reasonable certainty with respect to the boundaries of the claim term "sandbox."

The specification describes the sandboxes as being "for programmable electronic circuits" "for user-defined functionality." '753 patent, 5:19-21, 12:60-61, 3:13-14 ("for user-defined electronic circuits"), 4:29-30 ("for configurable modules").

However, in contrast, Plaintiffs' Infringement Contentions expand the scope of "sandbox" beyond PLDs to encompass "P4 engines" in an ASIC that does not have user-definable programmable circuits. '753 Infringement Contentions at 53. The intrinsic record does not provide any written description support for sandboxes that do not have user-definable programmable circuits. Therefore, the intrinsic record fails to inform a POSITA with reasonable certainty regarding the boundaries of "sandbox" under Plaintiffs' apparent view of the claim, particularly when Plaintiffs' Infringement Contentions appear to take an expansive view that a

sandbox could encompass any arbitrary portion of an integrated circuit, which renders the claims indefinite under 35 U.S.C. §112(b).

2. "memory superposition"

Under Plaintiffs' Infringement Contentions and apparent application of their interpretation of the claim language, claims 4, 7, 15, are invalid under 35 U.S.C. §112(b) because the claim language, when read in light of the intrinsic record, fails to inform a person of ordinary skill in the art the scope of the claimed invention with reasonable certainty, including with respect to the boundaries of what is included or not included in "memory superposition." For example, Plaintiffs Infringement Contentions allege this claim limitation is satisfied by pointing to the alleged presence of "various types of memory to optimize performance (e.g., various memory types like SRAM, TCAM, DRAM memory cache, DDR4/50," without addressing the claim's additional requirement of a selectable arrangement of the plurality of types of memory as multilevel cache, interleaved memory, swapped memory over a specified address range, and memory allocated to specified address ranges according to parameterized task needs. *See, e.g.*, '753 Infringement Contentions at 55. Therefore, Plaintiffs' Infringement Contentions and apparent application of their interpretation of the claim language renders the claims invalid for indefiniteness under 35 U.S.C. §112 because a POSITA would not be able to ascertain with reasonable certainty what is required for "memory superposition" or whether it can simply be satisfied by the presence of various types of memory as Plaintiffs allege.

3. "one or more swappable pods or cards" / "the plurality of swappable pods or cards"

Claims 1-4 of the '753 patent are invalid as indefinite under 35 U.S.C. §112 because the claim language, when read in light of the intrinsic record, fails to inform a POSITA of the scope of the claimed invention with reasonable certainty, including with respect to the claim terms

"one or more swappable pods or cards," and "the plurality of swappable pods or cards." For example, claim 1 "one or more swappable pods or cards" (which would encompass a single swappable pod or card) and then subsequently refers to "the plurality of swappable pods or cards" (which requires two or more swappable pods or cards). This internal inconsistency within the claims renders the claims invalid under §112 because the '753 patent fails to inform with reasonable certainty whether these claims can be satisfied by a single swappable pod or card.

D. U.S. Patent No. 10,985,943

1. "sandboxes"

Claims 1 and 10, and all claims depending from 1 and 10 recite "sandboxes." Under Plaintiffs' Infringement Contentions and apparent application of their interpretation of the claim language, these claims are invalid under 35 U.S.C. §112(b) because the claim language, when read in light of the intrinsic record, fails to inform a person of ordinary skill in the art the scope of the claimed invention with reasonable certainty with respect to the boundaries of the claim term "sandboxes."

The claims state that the sandboxes comprise part of a PLD and have "user-definable programmable electronic circuits of the PLD." '943 patent, claim 1 (limitation [1c]). The specification similarly describes the sandboxes as being "for programmable electronic circuits" "for user-defined functionality." '943 patent, 3:8-16, 12:54-65, 4:27-30 ("for user-defined electronic circuits"), 5:19-29 ("for configurable modules").

However, in contrast, Plaintiffs' Infringement Contentions expand the scope of "sandboxes" beyond PLDs to encompass portions ASICs that do not have user-definable programmable circuits of a PLD. The intrinsic record does not provide any written description support for sandboxes that do not have user-definable programmable circuits. Therefore, the intrinsic record fails to inform a POSITA with reasonable certainty regarding the boundaries of

"sandboxes" under Plaintiffs' apparent view of the claim, particularly when Plaintiffs' Infringement Contentions appear to take an expansive view that a sandbox could encompass any arbitrary portion of an integrated circuit, which renders the claims indefinite under 35 U.S.C. §112(b).

E. U.S. Patent No. 10,944,634

1. "The network connected device of claim 15, wherein the one or more processors are configurable to: send the parameter values for determination of bandwidth, from the first endpoint device, to the tuner server that is configured to receive further parameter values for determination of bandwidth of network connections from a plurality of endpoint devices; send, from the first endpoint device to the tuner server, connection information for the next network connection, wherein the tuner server performs the determining and derives one or more parameter values for the transmission bandwidth; and receive, at the first endpoint device, from the tuner server, the one or more parameter values for the transmission bandwidth."

The Asserted Claims of the '634 patent are invalid as indefinite under 35 U.S.C. § 112. Numerous terms and limitations in the Asserted Claims of the '634 patent, even when read in light of the specification and prosecution history, fail to inform with reasonable certainty one of ordinary skill in the art about the scope of the invention. For example, Asserted Claim 16 of the '634 patent requires: "The network connected device of claim 15, wherein the one or more processors are configurable to: send the parameter values for determination of bandwidth, from the first endpoint device, to the tuner server that is configured to receive further parameter values for determination of bandwidth of network connections from a plurality of endpoint devices; send, from the first endpoint device to the tuner server, connection information for the next network connection, wherein the tuner server performs the determining and derives one or more parameter values for the transmission bandwidth; and receive, at the first endpoint device, from the tuner server, the one or more parameter values for the transmission bandwidth." *See* '634 patent, 16:9-22. Limitations of claim 16 are not adequately defined in the patent specification,

and whose lack of reasonably ascertainable scope is compounded by Plaintiffs' overbroad and vague infringement contentions. Thus, at a minimum, Asserted Claim 16 of the '634 patent fails to distinctly claim the subject matter that the applicant regards as his invention, and are invalid under §112(b). *See Nautilus*, 572 U.S. 908-12.

Moreover, the Asserted Claims of the '634 patent are invalid for lack of written description. One of ordinary skill in the art at the time of the alleged invention would not have understood the applicant to be in possession of an invention comprising numerous of the limitations of the Asserted Claims. For example, the '634 patent specification fails to describe (such that one skilled in the art would have understood the applicant was in possession of) these claimed functionalities recited in the Asserted Claim 16 of the '634 patent.

2. "The network connected device of claim 15, further comprising the network connected device configurable to: monitor network traffic between the first endpoint device and the second endpoint device; transmit data associated with the network traffic from the first endpoint to the network connected device; determine, at the network connected device, an adjustment to the transmission bandwidth is desired; receive data for adjusting the transmission bandwidth from the network connected device; and adjust the transmission bandwidth responsive to the receiving."

As an additional example, Asserted Claim 20 of the '634 patent requires: "The network connected device of claim 15, further comprising the network connected device configurable to: monitor network traffic between the first endpoint device and the second endpoint device; transmit data associated with the network traffic from the first endpoint to the network connected device; determine, at the network connected device, an adjustment to the transmission bandwidth is desired; receive data for adjusting the transmission bandwidth from the network connected device; and adjust the transmission bandwidth responsive to the receiving." *See* '634 patent, 16:37-48. Limitations of claim 20 are not adequately defined in the patent specification, and whose lack of reasonably ascertainable scope is compounded by Plaintiffs' overbroad and vague

infringement contentions. Thus, at a minimum, Asserted Claim 20 of the ’634 patent fails to distinctly claim the subject matter that the applicant regards as his invention, and are invalid under §112(b). *See Nautilus*, 572 U.S. 908-12.

Moreover, the Asserted Claims of the ’634 patent are invalid for lack of written description. One of ordinary skill in the art at the time of the alleged invention would not have understood the applicant to be in possession of an invention comprising numerous of the limitations of the Asserted Claims. For example, the ’634 patent specification fails to describe (such that one skilled in the art would have understood the applicant was in possession of) these claimed functionalities recited in the Asserted Claim 20 of the ’634 patent.

3. “matches a geographical area of a past network connection”

One of ordinary skill in the art at the time of the alleged invention would not have understood the applicant to be in possession of an invention comprising numerous of the limitations of the Asserted Claims. For example, Asserted Claims 1, 8, and 15 of the ’634 patent require “matches a geographical area of a past network connection.” *See, e.g.*, ’634 patent, 14:7-8, 14:61-62, 16:1-2. The ’634 patent specification, however, fails to describe (such that one skilled in the art would have understood the applicant was in possession of) this claimed functionality.

4. “initiate/initiating the next network connection … with a transmission bandwidth based on the parameter values for the past network connection”

Additionally, Asserted Claims 1, 8, and 15 of the ’634 patent also require “initiate/initiating the next network connection … with a transmission bandwidth based on the parameter values for the past network connection.” *See, e.g.*, ’634 patent, 14:9-13, 14:63-67, 16:3-8. The ’634 patent specification, however, fails to describe (such that one skilled in the art would have understood the applicant was in possession of) this claimed functionality.

5. "wherein the first endpoint device is contained within the network connected device"

Asserted Claim 17 of the '634 patent requires "wherein the first endpoint device is contained within the network connected device." *See* '634 patent, 16:25-26. The '634 patent specification, however, fails to describe (such that one skilled in the art would have understood the applicant was in possession of) this claimed feature.

III. Identification of Claims Defendants Contend Are Directed to Ineligible Subject Matter Under 35 U.S.C. § 101

A. U.S. Patent No. 9,529,767

Defendants contend that all claims of the '767 patent are invalid under 35 U.S.C. §101 for claiming an abstract idea.

Under *Alice* step one, the Asserted Claims of the '767 patent are directed to an abstract idea. The '767 patent purports to solve a problem relating to the field of storage devices. Describing the then state-of-the-art, the '767 patent describes how a host (a computer) uses an interface protocol, such as Serial Advanced Technology Attachment (also known as Serial ATA or SATA) or Serial Attached SCSI (also known as SAS), to interface with a storage media device (e.g., hard drive or solid state drive). '767 pat. at 21-25. The '767 patent describes some of the differences between the SATA and SAS interface protocols. *Id.* at 1:25-38. The '767 patent observes that if a host needs to access a mixture of storage devices (e.g., some SATA drives and some SAS drives), "the host must support and manage both protocols," and that one consequence is that "[t]he host software layers interfacing to SATA and SAS must support their high protocol overhead." *Id.* at 1:36-42.

The '767 patent is directed toward its proposed solution, namely, to implement an abstraction protocol that is used by the host to communicate with storage controllers, which in

turn translate the abstract, high-level commands from the host into native commands understood by the storage device (e.g., SATA or SAS):

> This invention simplifies a host that accesses SATA and SAS storage media devices by abstracting the SATA and SAS protocols with one full duplex protocol that supports full command queuing to each storage media device, whether SATA or SAS. The invention is comprised of both hardware (system—architecture) and software (method—abstraction protocol). The simplified abstraction protocol provides high-level access commands to storage media devices.

Id. at 3:65-4:6.

> The storage media devices SATA or SAS interfaces are pushed to the edge, away from the host and toward the storage media device for which it supports. Pushing the SAS/SATA protocol to the edge simplifies the host/controller since it does not implement the legacy SAS/SATA protocol, it can do what fits its application best.

Id. at 4:10-16. However, the '767 patent's solution is directed to the abstract idea of translating commands or instructions from one format to another. *See Automation Middleware Solutions, Inc. v. Invensys Sys.*, Inc., Case No. 2:15-CV-898, 2017 WL 11573945 (E.D. Tex. Mar. 31, 2017) (finding claims "directed to the abstract idea of 'a middle translating layer'").

Using a translation layer to translate commands or instructions in a higher-level format to commands or instructions that are a lower-level format supported by a particular device is an abstract idea that has long been applied in the field of computing, which includes storage technology. For example, in an earlier era of computing, computer programs were written using the instruction set defined by the computer's manufacturer. Computers by different manufacturers often used different and incompatible instruction sets. Even different computers made by the same manufacturer used different and incompatible instructions sets. This led to the development of programming languages that operate on a more abstract basis, such as FORTRAN, COBOL, C, and C++. Instead of writing a computer program in the instruction set of a specific computer, the programmer writes the program in a given programming language. A compiler then is used to translate the program into the instruction set used by a particular

computer. Java programming is another application of this abstract idea. In Java, the programmer writes code in Java, and it is compiled into a bytecode that can be understood by a Java runtime environment. When the program is run on a computer, the Java runtime environment translates the bytecode into an instruction set used by the computer.

Under *Alice* step 2, the Asserted Claims of the '767 patent do not contain an inventive concept. The recited abstraction protocol is nothing more than the abstract idea itself. Instead, the abstraction protocol is merely described in functional terms, or in terms of objectives sought to be achieved from using an abstraction protocol:

> [T]he present invention is related to a system and method for abstracting SATA and/or SAS storage media devices via a full duplex queued command interface to increase performance, lower host over head, and simplify scaling storage media devices and system.

'767 pat. at 1:14-19.

> [T]he master controller and the edge controller communicate via the abstraction protocol, the abstraction protocol comprising a full-duplex protocol supporting full command queuing for the at least one storage media device.

Id. at 2:36-40.

> The abstraction protocol supports flow control between the host and the edge.

Id. at 4:30-32.

> This abstraction protocol is performance-centric and supports common high-level read and write access to a pool of storage media devices, each of which may have a SATA or SAS interface. The abstraction protocol is link-agnostic and may be carried via a multiplicity of direct attach or networked interfaces, including but not limited to PCIe, Ethernet (e.g., 1GbE, 10GbE, 40GbE, or 100GbE), Infiniband, USB, ThunderBolt, Firewire, and/or custom interfaces. The abstraction protocol link bandwidth need not be related to the bandwidth of the individual storage media devices SATA or SAS interface.

Id. at 4:35-45.

> The abstraction protocol supports packet-switched routable fabric and error recovery. Routing may be based on logical address, WWN (World Wide Number—a unique global ID number), or physical address. Logical address or WWN may be

> mapped to a physical address (e.g. by table lookup). This mapping may be static or dynamically updated based upon a discovery process as storage media devices are (i) inserted or removed from the system and/or (ii) powered on or off.

Id. at 4:55-63; *see also id.* at 3:65-4:6 and 4:10-16. The Federal Circuit has repeatedly stated that "a claim is abstract if it describes a process in 'result-based functional language' and fails to 'sufficiently describe how to achieve these results in a non-abstract way.'" *Voip-Pal.Com, Inc. v. Apple Inc.*, 411 F. Supp. 3d 926, 951 (N.D. Cal. 2019) (quoting *Two-Way Media Ltd. v. Comcast Cable Comm'cs, LLC*, 874 F.3d 1329, 1337 (Fed. Cir. 2017); *see also Bridge & Post, Inc. v. Verizon Commc'ns, Inc.*, 778 Fed. Appx. 882, 895 (Fed. Cir. 2019) (Bryson, CJ concurring) ("The distinction between claims that recite functions or results (the 'what it does' aspect of the invention) and those that recite concrete means for achieving particular functions or results (the 'how it does it' aspect of the invention) is an important indicator of whether a claim is directed to an abstract idea."). "Indeed, the essentially result-focused, functional character of claim language has been a frequent feature of claims held ineligible under § 101." *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1356 (Fed. Cir. 2016).

Notably, the specification and figures do not disclose a single example of the allegedly inventive abstraction protocol or provide clear guidance for a POSITA to implement the allegedly inventive abstraction protocol—highlighting that the abstraction protocol is little more than the abstract idea itself.

The other recited elements of the claim (e.g., packet-switched networks, hosts, master controllers, edge controllers, and storage media devices) do not provide an inventive concept. For example, the '767 patent does not purport to require any modifications to any packet-switched networks or storage media devices, which are otherwise well-understood, routine and conventional in the field of storage technology.

In the field of storage technology, a host (e.g., computer) was well-understood, routine and conventional. The '767 patent does not describe an inventive host, and the specification merely describes that the host uses the abstraction protocol to issue commands to storage media devices instead of using legacy protocols and commands (e.g., SATA or SAS). *Id*. at 3:65-4:6 and 4:10-16.

In the field of storage technology, storage controllers were well-understood, routine, and conventional. The '767 patent claims master controllers and edge controllers, but does not describe how they are inventive other than by functional description that they implement the abstraction protocol:

> A master controller 304, … , contains (i) one or more master processing elements 306 that *carries out the processing of the master side of the abstraction protocol* and communicates with at least one host 302 via a master controller interface 308 connected to the host 302 by a host link 320 and (ii) one or more control link interfaces 310 through 312, which communicate with one or more edge controllers 316 through 318 via control links 320 and 322.… Together, the host 302, the master controller 304, and edge controllers 316 through 318 form a routable packet-switched network supported by an abstraction protocol.

E.g., id. at 4:64-5:13 (emphasis added).

> The edge controller 402 contains (i) one or more edge processing elements 404 that *carries out the processing of the edge side of the abstraction protocol* and communicates with one or more master controllers 406 & 408 via control link interfaces 410 & 412 communicating with a master controllers 406 & 408 via control links 414 & 416: (ii) one or more SMD interfaces 418 & 420 which communicate with one or more SMDs 422 & 424 via SATA and/or SAS links 426 and 428; (iii) optionally, one or more forward link interfaces 430 communicating with other edge controllers 432 via control links 448 used to forward those abstract protocol packets from master controllers 406 or 408 (received via control link interfaces 410 & 412 that are not destined for the edge controller 402 that received them) to another edge controller 432; and (iv) optionally, one or more master controllers 434 & 436 communicating with other edge controllers 438 & 440 via control links 442 & 444 when an edge processing element 404 wishes to *author abstract protocol messages* to other edge controllers 438 & 440.

E.g., *id.* at 5:32-51 (emphasis added). Moreover, using a storage controller to translate commands from one protocol/format into a protocol/format compatible with a storage media

device was well-understood, routine, and conventional. *See, e.g.*, Ex. 767-05 (Malwankar) at limitations [1d] and [16e], Ex. 767-10 (Voorhees '224) at limitations [1d] and [16e].

B. U.S. Patent No. 10,944,634

Defendants contend that all claims of the '634 patent are invalid under 35 U.S.C. §101 for claiming an abstract idea. The basis for this contention is disclosed in Defendants' motion to dismiss for failure to state a claim under 35 U.S.C. § 101 for the '634 patent (Dkt. 23, 32), which is fully briefed and pending, and is hereby incorporated by reference. In addition, the prior art identified above for the '634 patent is further evidence that the claimed elements are well-understood, routine, or conventional.

IV. Production of Prior Art References in the Invalidity Contentions

The prior art referenced in these invalidity contentions have been produced to Plaintiffs, including in the documents bates labeled PA000001-PA040817.

Dated: December 20, 2024

/s/ Harper S. Batts

Harper S. Batts (CA #242603)
hbatts@sheppardmullin.com
Christopher Ponder (TX #24065916)
cponder@sheppardmullin.com
Jeffrey Liang (CA #281429)
jliang@sheppardmullin.com
Lai L. Yip (CA #258029)
lyip@sheppardmullin.com
SHEPPARD, MULLIN,
RICHTER & HAMPTON LLP
1540 El Camino Real, Suite 120
Menlo Park, CA 94025-4111
Tel.: (650) 815-2600
Fax: (650) 815-2601

Attorneys for Defendants
Advanced Micro Devices, Inc.
and Pensando Systems, Inc.

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of Defendants' Invalidity Contentions and Exhibits were served on Plaintiffs' counsel by email and Box.com shared folder sent to the following recipients on December 20, 2024:

Amy L. Ruhland (aruhland@reichmanjorgensen.com)

Ariane Mann (amann@reichmanjorgensen.com)

Christine E. Lehman (clehman@reichmanjorgensen.com)

Michael Matulewicz-Crowley (mmatulewicz-crowley@reichmanjorgensen.com)

Michelle Carr Quintana (mquintana@reichmanjorgensen.com)

Mira Yohannes (myohannes@reichmanjorgensen.com)

RJ_XtreamEdge-AMD Distribution List (rj_xtreamedge-amd@reichmanjorgensen.com)

Date: December 20, 2024

/s/ Harper S. Batts
Harper S. Batts