SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
HARPER S. BATTS, Cal. Bar No. 242603
hbatts@sheppardmullin.com
CHRISTOPHER S. PONDER, Cal. Bar No. 296546
cponder@sheppardmullin.com
JEFFREY LIANG, Cal. Bar No. 281429
jliang@sheppardmullin.com
LAI L. YIP, Cal. Bar No. 258029
lyip@sheppardmullin.com
1540 El Camino Real Suite 120
Menlo Park, California 94025-4111
Telephone: 650.815.2600

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

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA, OAKLAND DIVISION

CONCURRENT VENTURES, LLC AND XTREAMEDGE, INC., Plaintiffs, v. ADVANCED MICRO DEVICES, INC. AND PENSANDO SYSTEMS, INC., Defendants.	Case No. 4:25-cv-09567-JST **DEFENDANTS ADVANCED MICRO DEVICES, INC. AND PENSANDO SYSTEMS, INC.'S INVALIDITY CONTENTIONS AND ACCOMPANYING DOCUMENT PRODUCTION**

Pursuant to the Scheduling Order (Dkt. 134), Defendants Advanced Micro Devices, Inc. and Pensando Systems, Inc. hereby disclose their Invalidity Contentions and Accompanying Document Production (Pat. L.R. 3-3 and 3-4).

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 Plaintiffs' 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. Patent L.R. 3-3(A)-(C): Identification of Prior Art That Anticipates or Renders Obvious Each Asserted Claim

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 their 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 their 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
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

Prior Art to ’596	Anticipation/ Obviousness	Claim Chart
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), including but not limited to TCP/IP RTOS, ARM1156T2-S, Cortex-R5, Cortex-A, ARMv7, Corelink CCI-400, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. ARM System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. ARM System(s) were publicly used by ARM, the public, and third-party customers before this date, by December 5, 2013 at the latest.	A & O	Exhibit 596-18

Prior Art to ’596	Anticipation/ Obviousness	Claim Chart
Intel/Fulcrum System(s), including but not limited to FM4000 24-Port 10G Ethernet L2/L3/L4 Switch/Router, FM4000 Series 10GbE Low Latency L2/L3 Switching Silicon, FocalPoint FM4000 Series 10G Enhanced Ethernet Switch-Router, FocalPointII A 240 Gb/s switch router, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. Intel Fulcrum System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. Intel/Fulcrum System(s) were publicly used by Intel/Fulcrum, the public, and third-party customers before this date, by December 5, 2013 at the latest.	A & O	Exhibit 596-19
NVIDIA/Mellanox System(s) including but not limited to RDMA Aware Networks, ConnectX-2 Single/Dual-Port Adapter Device with Virtual Protocol Interconnect, Mellanox OFED for Linux, Mellanox Technologies InfiniBand Fabrics, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. NVIDIA/Mellanox System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. NVIDIA/Mellanox System(s) were publicly used by the NVIDIA/Mellanox, the public, and third-party customers before this date, by December 5, 2013 at the latest.	A & O	Exhibit 596-20
U.S. Patent Application Publication No. 2005/0071529 to Borkowski et al. (“Borkowski”); filed on September 30, 2003; published on March 31, 2005	O	Exhibit 596-21
U.S. Patent No. 7,844,758 to Hughes (“Hughes”); filed on June 18, 2003; issued on November 30, 2010	A & O	Exhibit 596-22
U.S. Patent No. 6,938,253 to Kim (“Kim”); filed on May 2, 2001; issued on August 30, 2005	O	Exhibit 596-23

Prior Art to ’596	Anticipation/ Obviousness	Claim Chart
U.S. Patent Application Publication No. 2006/0136640 to Tuan (“Tuan”); filed on April 28, 2005; published on June 22, 2006	O	Exhibit 596-24
IBM Cell System(s) including but not limited to IBM Cell Broadband Engine were sold, offered for sale, made known, and/or publicly used before December 6, 2013. IBM Cell System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. IBM Cell System(s) were publicly used by IBM, the public, and third-party customers before this date, by December 5, 2013 at the latest.	A & O	Exhibit 596-25

In addition to the anticipation and obviousness explained in this section, the ’596 patent is anticipated and/or rendered obvious by prior art as explained by Defendants’ Petition for Inter Partes Review (assigned Case No. IPR2025-00478), which is incorporated herein by reference:

Grounds	Claims	Prior Art
1	1-2, 5-8, 11-14, 17-18	Dongare in view of Gewirtz
2	3-4, 9-10, 15-16	Ground 1 further in view of Matsunami
3	1-2, 6-8, 12-14, 18	Khawand in view of Rosenthal
4	1-2, 6-8, 12-14, 18	Ground 3 further in view of Ohkawa
5	3-4, 9-10, 15-16	Ground 4 further in view of Matsunami

1. Anticipation

For each item of prior art identified above as an anticipation reference, Defendants contend that the prior art item anticipates the one or more 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**
- **Alexander + Borkowski**
- **Alexander + Hughes**
- **Alexander + Kim**
- **Alexander + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Alexander with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Alexander with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Alexander with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Alexander with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Alexander, and a POSITA would have been motivated to combine for similar reasons.

- **Buchko + Drews**
- **Buchko + Gewirtz**
- **Buchko + Goyal**
- **Buchko + Kolinummi**
- **Buchko + Ohkawa**
- **Buchko + Rosenthal**
- **Buchko + Burkowski**
- **Buchko + Hughes**
- **Buchko + Kim**
- **Buchko + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Buchko with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Buchko with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Buchko with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Buchko with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Buchko, and a POSITA would have been motivated to combine for similar reasons.

- **Chavan + Drews**
- **Chavan + Gewirtz**

- **Chavan + Goyal**
- **Chavan + Kolinummi**
- **Chavan + Ohkawa**
- **Chavan + Rosenthal**
- **Chavan + Borkowski**
- **Chavan + Hughes**
- **Chavan + Kim**
- **Chavan + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Chavan with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Chavan with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Chavan with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Chavan with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Chavan, and a POSITA would have been motivated to combine for similar reasons.

- **Dongare + Drews**
- **Dongare + Gewirtz**
- **Dongare + Goyal**
- **Dongare + Kolinummi**
- **Dongare + Ohkawa**
- **Dongare + Rosenthal**
- **Dongare + Borkowski**
- **Dongare + Hughes**
- **Dongare + Kim**
- **Dongare + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Dongare with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Dongare with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Dongare with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Dongare with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Dongare, and a POSITA would have been motivated to combine for similar reasons.

- **Hammer + Drews**
- **Hammer + Gewirtz**
- **Hammer + Goyal**
- **Hammer + Kolinummi**
- **Hammer + Ohkawa**
- **Hammer + Rosenthal**
- **Hammer + Borkowski**
- **Hammer + Hughes**
- **Hammer + Kim**
- **Hammer + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Hammer

with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Hammer with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Hammer with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Hammer with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Hammer, and a POSITA would have been motivated to combine for similar reasons.

- **Khawand + Drews**
- **Khawand + Gewirtz**
- **Khawand + Goyal**
- **Khawand + Kolinummi**
- **Khawand + Ohkawa**

- **Khawand + Rosenthal**
- **Khawand + Borkowski**
- **Khawand + Hughes**
- **Khawand + Kim**
- **Khawand + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Khawand with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Khawand with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Khawand with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Khawand with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Khawand, and a POSITA would have been motivated to combine for similar reasons.

- **Mannava + Drews**

- **Mannava + Gewirtz**
- **Mannava + Goyal**
- **Mannava + Kolinummi**
- **Mannava + Ohkawa**
- **Mannava + Rosenthal**
- **Mannava + Borkowski**
- **Mannava + Hughes**
- **Mannava + Kim**
- **Mannava + Tuan**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

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. As an additional example, a POSITA would have been motivated to combine Mannava with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Mannava with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Mannava with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Mannava with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Mannava, and a POSITA would have been motivated to combine for similar reasons.

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

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. As an additional example, a POSITA would have been motivated to combine Michaylov with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Michaylov with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Michaylov with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Michaylov with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Michaylov, and a POSITA would have been motivated to combine for similar reasons.

- **Van Loo + Drews**
- **Van Loo + Gewirtz**
- **Van Loo + Goyal**
- **Van Loo + Kolinummi**
- **Van Loo + Ohkawa**
- **Van Loo + Rosenthal**
- **Van Loo + Borkowski**
- **Van Loo + Hughes**
- **Van Loo + Kim**
- **Van Loo + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Van Loo with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Van Loo with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Van Loo with Kim, which

teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Van Loo with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Van Loo, and a POSITA would have been motivated to combine for similar reasons.

- **ARM System(s) + Drews**
- **ARM System(s) + Gewirtz**
- **ARM System(s) + Goyal**
- **ARM System(s) + Kolinummi**
- **ARM System(s) + Ohkawa**
- **ARM System(s) + Rosenthal**
- **ARM System(s) + Borkowski**
- **ARM System(s) + Hughes**

- **ARM System(s) + Kim**
- **ARM System(s) + Tuan**
- **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. As an additional example,

a POSITA would have been motivated to combine ARM System(s) with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine ARM System(s) with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine ARM System(s) with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to the ARM System(s), and a POSITA would have been motivated to combine for similar reasons.

- **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**
- **Intel/Fulcrum System(s) + Borkowski**
- **Intel/Fulcrum System(s) + Hughes**
- **Intel/Fulcrum System(s) + Kim**
- **Intel/Fulcrum System(s) + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Intel/Fulcrum System(s) with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Intel/Fulcrum System(s) with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to the Intel/Fulcrum System(s), and a POSITA would have been motivated to combine for similar reasons.

- **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**
- **Nvidia/Mellanox System(s) + Borkowski**
- **Nvidia/Mellanox System(s) + Hughes**
- **Nvidia/Mellanox System(s) + Kim**
- **Nvidia/Mellanox System(s) + Tuan**
- **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. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motived to combine Nvidia/Mellanox System(s) with Hughes, which teaches a queue reservation system with a free entry counter. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s), which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Nvidia/Mellanox System(s) with Tuan, which

teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to the Nvidia/Mellanox System(s), and a POSITA would have been motivated to combine for similar reasons.

- **Hughes + Drews**
- **Hughes + Gewirtz**
- **Hughes + Goyal**
- **Hughes + Kolinummi**
- **Hughes + Ohkawa**
- **Hughes + Rosenthal**
- **Hughes + Borkowski**
- **Hughes + Hughes**
- **Hughes + Kim**
- **Hughes + Tuan**
- **Hughes + Kanai**

- **Hughes + Matsunami**

In addition to the above-mentioned motivations to combine, Hughes discloses a multiprocessor computing system in which "multiple sources of transactions" "share a common queuing structure which services those transactions," including shared request queues used to service requests to other processing nodes and system memory. Background; 1:43-50. Hughes further teaches that the request queues are "responsible for accepting an . . . request . . . performing actions necessary for completing the request, and returning appropriate responses back to the requestors as necessary." Background. Hughes explains that queue "complexity" can make "each queue entry" consume "significant area," so "limiting the total number of queue entries may be required," creating a need for "an efficient method of allocating and managing the available queue resources." Hughes, Background. To address this, Hughes teaches a request queue management technique that keeps track of the number of free queue entries for each requestor processor and, at initialization, "allots to each requestor" predetermined "hard" and "free" entries, with remaining "un-allotted entries" forming "a free pool of entries." Hughes, Abstract; Summary.

Therefore, given the context provided by Hughes, 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 Hughes 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 Hughes 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 Hughes 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 Hughes 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 Hughes 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 Hughes 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. As an additional example, a POSITA would have been motivated to combine Hughes with Borkowski, which teaches the use of counting semaphores for use in resource-locking including FIFOs. As an additional example, a POSITA would have been motivated to combine Hughes with Kim, which teaches a multiprocessing architecture that enables inter-processor communication using a set of hardware semaphore registers. As an additional example, a POSITA would have been motivated to combine Hughes with Tuan, which teaches a multiprocessor system that uses hardware semaphores to arbitrate access to shared resources. A POSITA would naturally have looked to these references and been motivated to combine them with Hughes 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. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

Additionally, Hughes discloses support for I/O operations interfacing with external devices such as non-volatile memory devices. 1:10–22; 2:37–63. 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 Hughes 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. The motivations to combine explained in the incorporated Petition (IPR2025-00487) also apply to Hughes, and a POSITA would have been motivated to combine for similar reasons.

- **IBM Cell Broadband System(s) + Alexander**
- **IBM Cell Broadband System(s) + Buchko**
- **IBM Cell Broadband System(s) + Dongare**
- **IBM Cell Broadband System(s) + Drews**

- **IBM Cell Broadband System(s) + Gewirtz**
- **IBM Cell Broadband System(s) + Goyal**
- **IBM Cell Broadband System(s) + Hammer**
- **IBM Cell Broadband System(s) + Khawand**
- **IBM Cell Broadband System(s) + Kolinummi**
- **IBM Cell Broadband System(s) + Michaylov**
- **IBM Cell Broadband System(s) + Ohkawa**
- **IBM Cell Broadband System(s) + Van Loo**
- **IBM Cell Broadband System(s) + Borkowski**
- **IBM Cell Broadband System(s) + Hughes**
- **IBM Cell Broadband System(s) + Kim**
- **IBM Cell Broadband System(s) + Tuan**
- **Any of the above + Kanai**
- **Any of the above + Matsunami**

IBM Cell Broadband System(s) teaches a multiprocessor architecture with Power Processing Engines ("PPEs") and Synergistic Processing Units ("SPUs") that communicate with each other using hardware queues managed by a Memory Flow Control Unit ("MFCU"). As explained in the accompanying claim chart, IBM Cell Broadband System(s) teaches queue management, and there were teachings, suggestions, and motivations in the art for IBM Cell Broadband System(s) for implementing queue management functionality, *e.g.*, for the MFCU, using computer instructions running on a processor. *See, e.g.*, Day at PA043552-PA043628 at 11:25-42. A POSITA would therefore have been motivated to combine IBM Cell Broadband System(s) with known queue management techniques that use software instructions on processors, particularly prior art for multiprocessor systems and flow control. For example, a POSITA would have been motivated to combine IBM Cell Broadband System(s) with Michaylov, which teaches techniques for using software instructions, running on a processor, for queue management. Michaylov, 13:9-44 ("The queue management functions described herein can be implemented using software for execution on a computer."). As other examples, a POSITA would have been

motivated to combine IBM Cell Broadband System(s) with Alexander, Buchko, Dongare, Drews, Gewirtz, Goyal, Hammer, Khawand, Kolinummi, Ohkawa, Van Loo, Borkowski, Hughes, Kim, and/or Tuan, which likewise taught using software instructions, running on a processor, for queue management. A POSITA would naturally have looked to these references and been motivated to combine them with IBM Cell Broadband System(s), for numerous reasons, to implement queue management functionality, e.g., for the MFCU, using computer instructions running on a processor. First, the Cell MFCU, Michaylov, Alexander, Buchko, Dongare, Drews, Gewirtz, Goyal, Hammer, Khawand, Kolinummi, Ohkawa, Van Loo, Borkowski, Hughes, Kim, and Tuan all manage queues in a multiprocessor system and/or for staging requests for memory operation. Therefore, a POSITA looking at the operation of queue management in the Cell architecture, including MFCU operation, would have been motivated to combine similar solutions from other multiprocessor systems, such as Michaylov and the other listed references, for the same function taught by each reference. Second, the Cell MFCU was highly configurable, allowing flexible configuration of its operations. Therefore, a POSITA would have been motivated to increase configurability and flexibility using software running on a processor to implement MFCU functionality in IBM Cell Broadband System(s). Michaylov, Alexander, Buchko, Dongare, Drews, Gewirtz, Goyal, Hammer, Khawand, Kolinummi, Ohkawa, Van Loo, Borkowski, Hughes, Kim, and Tuan teach similar queue management functionality implemented using software running on a processor. Therefore, a POSITA would have been motivated to combine these references with IBM Cell Broadband System(s). Third, a POSITA would have been motivated by several advantages of the above-described combinations, including a simplified implementation for providing increased configurability, flexibility, and user-customizability. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results.

IBM Cell Broadband System(s) include a flow control register (e.g., queue status register) that indicates the available space in a queue, and it teaches checking this register before writing to its corresponding queue, to ensure there is sufficient space. After a write, the status register would

need to be updated to reflect the new amount of available space remaining. Therefore, a POSITA would have been motivated to combine IBM Cell Broadband System(s) with known techniques for flow control in multiprocessor systems, for updating status registers. Tuan teaches this by decrementing the value stored at a flow control register before returning the result of a read to that register. A POSITA would have been motivated to combine these teachings because they address similar flow-control issues in multiprocessor systems, and the combination would have been a straightforward application and use of a known technique to a known device ready for improvement to yield predictable results. Applying these techniques also would have been obvious to try, with a reasonable expectation of success, because there were only a finite number of identified, predictable, and known solutions with respect to ordering the steps of decrementing and returning. Indeed, a POSITA would have been motivated because the combination was a simple way to update the flow control register, without separate synchronization or additional write operations.

Additionally, IBM Cell Broadband System(s) disclose the support of I/O operations, with various I/O ports for a variety of external devices. 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 IBM Cell Broadband 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. Moreover, IBM Cell Broadband System(s) were designed to be used in a wide variety of applications, including for example in servers, high-performance computing systems, and even video game consoles. Its architecture was designed to process streams of data. Therefore, a POSITA would have been motivated to use IBM Cell Broadband System(s) with SAN and NAS because both operate on streams of data in a network environment. This would have been simple combination of known prior art element according to known methods to yield predictable results as well as a simple substitution of known elements for another to obtain predictable results. The motivations to combine explained in the

incorporated Petition (IPR2025-00487) also apply to the IBM Cell Broadband System(s) and a 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. IPR2025-00487).

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(s), including but not limited to EMC VPLEX VS1, EMC VPLEX VS2, EMC VPLEX Local, EMC VPLEX Metro, EMC VPLEX Geo, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. EMC VPLEX System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. EMC VPLEX System(s) were publicly used by Dell/EMC, the public, and third-party customers before this date, by December 5, 2013 at the latest.	A & O	Exhibit 767-01
IBM SAN Volume Controller System(s), including but not limited to IBM SAN Volume Controller 2145-4F2, IBM SAN Volume Controller 2145-CF8, IBM SAN Volume Controller 2145-8F2, IBM SAN Volume Controller 2145-8F4, IBM SAN Volume Controller 2145-8G4, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. IBM SAN Volume Controller System(s) were sold, offered for sale, and/or	A & O	Exhibit 767-02

Prior Art to ’767	Anticipation/ Obviousness	Claim Chart
made known to the public and third-party customers before this date, by December 5, 2013 at the latest. IBM SAN Volume Controller System(s) were publicly used by IBM, the public, and third-party customers before this date, by December 5, 2013 at the latest		
Microsoft Server System(s), including but not limited to Microsoft Server 2012, Microsoft Server 2012 R2, Microsoft HPC Server 2008, Microsoft HPC Server 2008 R2, were sold, offered for sale, made known, and/or publicly used before December 6, 2013. Microsoft Server System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by December 5, 2013 at the latest. Microsoft Server System(s)were publicly used by Microsoft, the public, and third-party customers before this date, by December 5, 2013 at the latest	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
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

Prior Art to ’767	Anticipation/ Obviousness	Claim Chart
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
U.S. Patent No. 6,996,741 to Pittelkow et al. (“Pittelkow ’741”); filed on June 28, 2002; issued on December 27, 2006	A & O	Exhibit 767-13

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 + Pittelkow**

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

Northcutt discloses a scalable storage system that employs an "advanced storage system" to represent physical storage devices as virtual drives. *See* Exhibit 767-04. The advanced storage system acts as an intermediary between the host and the storage devices, connected in a cascaded topology. *See* Northcutt, [0021], [0023], [0030]. 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 (*See* Malwankar, 11:60-12:3, 14:38-49, 15:52-61, 15:62-16:15) Pittelkow (*see* Pittelkow, 20:11-26, 12:56-13:4, 13:6-35), Stephens (*see* Stephens, [0100], [0116], [0121], [0129]) or Steinmetz (*see* Steinmetz, 22:26-60, 23:23-32, 54:21-41), 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**
- **Northcutt + Pittelkow + Yendigiri**
- **Northcutt + Pittelkow + 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 (*see* Yendigiri, 2:19-33, 3:39-44, 4:41-43, 6:24-32) or Voorhees '224 (*see* Voorhees '224, 2:50-64, 8:63-9:11), 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.

The combination of Northcutt and Pittelkow similarly discloses a data storage system that supports various storage devices, including SATA devices and legacy SCSI hard drives. Therefore, for the same reason as above, a POSITA would have been motivated to further combine Northcutt and Pittelkow 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**
- **Northcutt + Pittelkow + Yendigiri + Johnson + Voorhees '096**
- **Northcutt + Pittelkow + 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. *See* Voorhees '096, Fig. 2, 3:4-46, 6:61-7:6, 8:4-16, 8:17-30. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. *See* Johnson, Fig. 4, [0017], [0022], [0023]. 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.

A POSITA would have been motivated to further combine Northcutt, Pittelkow and Yendigiri with Voorhees ’096 and Johnson, because of same reason explained above. Furthermore, because the combination of Northcutt, Pittelkow, and Yendigiri already establishes a functional framework for device discovery within a SAN architecture and explicitly supports SAS devices, it would have been a routine and obvious to incorporate the specific SAS discovery mechanisms taught by Voorhees ’096 and Johnson. Combining these references effectively “connects the dots” by implementing SAS-specific discovery within a SAN environment that is already configured for automated device discovery. For the same reasons, a POSITA would have been motivated to further combine Northcutt, Pittelkow and Voorhees ’224 with Voorhees ’096 and Johnson.

- **Malwankar + Klughart**
- **Malwankar + Pittelkow**

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. *See* Exhibit 767-05. It teaches a plurality of storage controllers and assigning storage devices to one or more controllers. *See* Malwankar, 1:13-47, 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. *See* Klughart, [0055], [0105], [0107], [0164]. Similarly, a POSITA would have been motivated to combine Malwankar with Pittelkow, which also resides in the field of storage area networks and teaches methods for inter-controller communication, including over the Ethernet channel. *See* Pittelkow, 4:9-31, 12:56-13:4, 13:6-35, 20:11-26.

- **Malwankar + Yendigiri**
- **Malwankar + Steinmetz**
- **Malwankar + Voorhees '224**
- **Malwankar + Klughart + Yendigiri**
- **Malwankar + Klughart + Steinmetz**
- **Malwankar + Klughart +Voorhees '224**
- **Malwankar + Pittelkow + Steinmetz**
- **Malwankar + Pittelkow + Voorhees '224**

Additionally, Malwankar teaches converting storage-based requests into a format compatible with the target device. *See* Malwankar, 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. *See* Voorhees '224, 2:50-64, 8:63-9:11. Similarly, a POSITA would have been motivated to combine Malwankar with Yendigiri (*see* Yendigiri, 2:19-33, 3:39-44, 4:41-43, 6:24-32) or Steinmetz (*see* Steinmetz, 53:36-49, 53:64-54:3), 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. Likewise, it would have been obvious to a POSITA to integrate the combined disclosures of Malwankar and Pittelkow 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**
- **Malwankar + Pittelkow + Yendigiri + Johnson + Voorhees ’096**
- **Malwankar + Pittelkow + Steinmetz + Johnson + Voorhees ’096**
- **Malwankar + Pittelkow + 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. *See* Voorhees ’096, Fig. 2, 3:4-46, 6:61-7:6, 8:4-16, 8:17-30. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. *See* Johnson, Fig. 4, [0017], [0022], [0023]. 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.

A POSITA would have been motivated to further combine Malwankar, Pittelkow and Yendigiri with Voorhees '096 and Johnson, because of same reason explained above. Furthermore, because the combination of Malwankar, Pittelkow, and Yendigiri already establishes a functional framework for device discovery within a SAN architecture and explicitly supports SAS devices, it would have been a routine and obvious to incorporate the specific SAS discovery mechanisms taught by Voorhees '096 and Johnson. Combining these references effectively "connects the dots" by implementing SAS-specific discovery within a SAN environment that is already configured for automated device discovery. For the same reason, a POSITA would have been motivated to further combine Malwankar, Pittelkow and Voorhees '224 with Voorhees '096 and Johnson, and combine Malwankar, Pittelkow, Steinmetz with Voorhees '096 and Johnson.

- **Pittelkow + Voorhees '224**
- **Pittelkow + Yendigiri**
- **Pittelkow + Steinmetz**

Pittelkow teaches a storage network implemented on the SAN fabric. *See* Exhibit 767-13. A POSITA would have recognized that the SAN fabric architecture taught by Pittelkow is media-agnostic and not limited to legacy parallel SCSI drives, but is equally capable of supporting other storage devices, including SAS storage devices. Given the industry-wide transition from parallel to serial storage protocols, a POSITA would have been naturally motivated to look to references providing implementation details for these modern drive types. For example, a POSITA would have been motivated to combine Pittelkow with Voorhees '224, which discloses converting host requests into SAS requests. *See* Voorhees '224, 2:50-64, 8:63-9:11. Similarly, a POSITA would have been motivated to combine Pittelkow with Yendigiri (*see* Yendigiri, 2:19-33, 3:39-44, 4:41-43, 6:24-32) or Steinmetz (*see* Steinmetz, 53:36-49, 53:64-54:3), each of which is directed to SAS devices.

- **Pittelkow + Klughart**
- **Pittelkow + Northcutt**
- **Pittelkow + Malwankar**

- **Pittelkow + Stephens**

Pittelkow teaches a system involving storage controllers. A POSITA would have found it obvious to implement the prior art's teachings relevant to the claimed master controller(s) or slave controller(s) using ASICs or FPGAs, well-known techniques as shown in Northcutt (*see* Northcutt, [0205]), Malwankar (*see* Malwankar, [0171]), Stephens (*see* Stephens, [0129]), and Klughart (*see* Klughart, [0059], [0259], [0261]). 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). Similarly, a POSITA would have found it obvious to use an FPGA to achieve similar hardware-level acceleration while maintaining the flexibility to reconfigure the controller logic.

- **Pittelkow + Voorhees '224**
- **Pittelkow + Yendigiri**
- **Pittelkow + Steinmetz**
- **Pittelkow + Voorhees '224 + Northcutt**
- **Pittelkow + Voorhees '224 + Klughart**
- **Pittelkow + Yendigiri + Northcutt**
- **Pittelkow + Yendigiri + Klughart**
- **Pittelkow + Steinmetz + Northcutt**
- **Pittelkow + Steinmetz + Klughart**

The combination of Pittelkow and Voorhees '224 establishes a scalable data storage architecture supporting SAS devices. Within such a networked environment, seamless communication between storage controllers is required to coordinate complex storage-based operations. While Pittelkow discloses several methods for inter-controller communication, it does not explicitly detail a mechanism for decomposing high-overhead host commands into smaller,

distributed tasks that can be processed across multiple controllers. *See* Pittelkow, 4:9-31, 12:56-13:4, 13:6-35, 20:11-26. A POSITA would have recognized that the performance of a distributed system depends on efficient command execution and the ability to scale processing across the controller fabric. 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 controllers. For example, a POSITA would have been motivated to further combine Pittelkow and Voorhees '224 with Northcutt (*see* Northcutt, [0004], [0018], [0025], [0031], [0036]) or Klughart (*see* Klughart, [0105], [0107], [0139]), each of which teaches one command transforming into more commands, to enhance the storage network system's capability of managing distributed tasks efficiently. Similarly, a POSITA would have been motivated to further combine Pittelkow and Steinmetz with Northcutt or Klughart, and further combine Pittelkow and Yendigiri with Northcutt or Klughart.

- **Pittelkow + Yendigiri + Johnson + Voorhees '096**
- **Pittelkow + Steinmetz + Johnson + Voorhees '096**
- **Pittelkow + Voorhees '224 + Johnson + Voorhees '096**

The combination of Pittelkow and Yendigiri establishes a storage network architecture that supports SAS devices and implements automated device discovery within a SAN environment. Because Pittelkow already discloses the framework for general SAN discovery, a POSITA would have found it obvious to incorporate the specific SAS discovery mechanisms required to manage a SAS-based topology. Therefore, a POSITA seeking to optimize network initialization and management would have naturally looked to specialized references disclosing SAS-specific discovery protocols. 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. *See* Voorhees '096, Fig. 2, 3:4-46, 6:61-7:6, 8:4-16, 8:17-30. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. *See* Johnson, Fig. 4, [0017], [0022], [0023]. Therefore, a POSITA would have been motivated to further combine Pittelkow

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 Pittelkow and Steinmetz with Voorhees ’096 and Johnson, further combine Pittelkow 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 (*see* Yendigiri, 2:19-33, 3:39-44, 4:41-43, 6:24-32), Steinmetz (*see* Steinmetz, 53:36-49, 53:64-54:3), or Voorhees ’224 (*see* Voorhees ’224, 2:50-64, 8:63-9:11), 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. *See* Voorhees ’096, Fig. 2, 3:4-46, 6:61-7:6, 8:4-16, 8:17-30. Johnson discloses performing a device discovery upon detecting a status change in the SAS topology and identifies events that could trigger such changes. *See* Johnson, Fig. 4, [0017], [0022], [0023]. 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 (*see* Northcutt, [0004], [0018], [0025], [0031], [0036]) or Klughart (*see* Klughart, [0105], [0107], [0139]), 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 (*see* Northcutt, [0004], [0018], [0025], [0031], [0036]) or Klughart (*see* Klughart, [0105], [0107], [0139]), 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 or FPGAs, well-known techniques as shown in Northcutt (*see* Northcutt, [0205]), Malwankar (*see* Malwankar, [0171]), Stephens (*see* Stephens, [0129]), and Klughart (*see* Klughart, [0059], [0259], [0261]). 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). Similarly, a POSITA would have found it obvious to use an FPGA to achieve similar hardware-level acceleration while maintaining the flexibility to reconfigure the controller logic.

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), including but not limited to Cray XC40, Cray XD1, and Cray XE, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Cray Supercomputer System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Cray Supercomputer System(s) were publicly used by HPE/Cray, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 753-01
BreakingPoint Scalable System(s), including but not limited to BreakingPoint Enterprise, BreakingPoint FireStorm, BreakingPoint Elite, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. BreakingPoint Scalable System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. BreakingPoint Scalable System(s) were publicly used by BreakingPoint, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 753-02
SeaMicro Computing System(s), including but not limited to SeaMicro SM15000 Cloud Servers, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. SeaMicro Computing System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. SeaMicro Computing System(s) were publicly used by SeaMicro/AMD, the public, and third-party customers before this date, by July 14, 2018 at the latest.	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

Prior Art to '753	Anticipation/ Obviousness	Claim Chart
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
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* (Jones & Bartlett Publishers 1996).	O	Exhibit 753-08
Internet Protocol, Request for Comments (RFC) 791 (Jon Postel ed., Sept. 1981), https://www.rfc-editor.org/rfc/rfc791.	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
Dell PC System(s), including but not limited to Dell XPS 13 Notebook and Dell Precision 5820 Tower, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Dell PC System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Dell PC System(s) were publicly used by Dell, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 753-12

Prior Art to ’753	Anticipation/ Obviousness	Claim Chart
IBM System(s), including but not limited to IBM z13, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. IBM System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. IBM System(s) were publicly used by IBM, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 753-13
U.S. Patent No. 9,317,429 to Ramanujan et al. (“Ramanujan”); filed September 30, 2011; issued April 19, 2016	O	Exhibit 753-14
Intel Optane System(s), including non-volatile memory devices and associated reference systems, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Intel Optane System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Intel Optane System(s) were publicly used by Intel, the public, and third-party customers before this date, by July 14, 2018 at the latest.	O	

In addition to the anticipation and obviousness explained in this section, the ’753 patent is anticipated and/or rendered obvious by prior art as explained by Defendants’ Petition for Inter Partes Review (assigned Case No. IPR2025-00486), which is incorporated herein by reference:

Grounds	Claims	Prior Art
1	1-3, 13, 14	Stroud and DaSilva
2	7, 9-10	Stroud and Cragon
3	7, 9-10	Stroud, Cragon, and Saulsbury
4	4, 8, 15	Stroud, DaSilva, Cragon, and Saulsbury

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**
- **Cray Supercomputer System(s) + RFC 791**
- **Cray Supercomputer System(s) + Da Silva**
- **Cray Supercomputer System(s) + Cragon**
- **Cray Supercomputer System(s) + Ramanujan**
- **Cray Supercomputer System(s) + Dell PC System(s)**
- **Cray Supercomputer System(s) + IBM System(s)**
- **Cray Supercomputer System(s) + Intel Optane System(s)**
- **Any of the above + Cullison or Saulsbury**

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, such as the use of a chassis identifier, module identifier, board identifier,

instance identifier, and type identifier in packet headers. 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 their teachings—including details and features for chassis, cards, and networks in data centers, such as the use of a chassis identifier, module identifier, board identifier, instance identifier, and type identifier in packet headers—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.

A POSITA would have similarly been motivated to combine Cray Supercomputer System(s) with RFC 791. Cray Supercomputer System(s) discloses a supercomputer architecture that sent packet communications throughout a network, which Cray Supercomputer System(s) teaches or suggests is an IP network over Ethernet. *See, e.g.,* Exhibit 753-01. RFC 791 is the widely-known standard governing the IP protocol. Therefore, a POSITA would have known to

use RFC 791 and been motivated to use the IP protocol with Cray Supercomputer System(s). For decades, the IP protocol has been ubiquitous in computers and computer networks—including for supercomputers such as Cray Supercomputer System(s). *See* RFC 791 at 1-3, 5-6, 8-9, and 11-23. Therefore, a POSITA would have looked to RFC 791 and been motivated to combine it with Cray Supercomputer System(s) packet-based network communications to facilitate packet-based communications over IP and Ethernet.

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 Ramanujan, which teaches a memory hierarchy and different configurable modes of memory operation. *See* Ramanujan at 3:32-35, 7:45-54, 8:11-29, 8:30-37, 9:1-10, 9:27-31, 9:34-40, 10:31-38, 13:3-9, 17:59-18:6, 31:5-12, 31:13-19, 31:29-32, FIG. 1, FIG. 2, FIG. 5B, FIG. 6D. Likewise, Cragon teaches multiple memory arrangements and modes of memory operation. *See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491. A POSITA would have also been motivated to look at the memory arrangements of existing computer systems at the time, including Dell PC System(s), which teaches multiple memory arrangements, including for DDR3/DDR4; L1, L2, and L3 cache; flash memory; hard disks; etc. (*see* Exhibit 753-12) and IBM System(s), which teaches multiple memory arrangements, including DDR3; L1, L2, L3, and L4 cache; flash memory; hard disks; etc. for performing analysis of large volumes of data and operational analytics of workloads throughout a cloud network (*see* Exhibit 753-13). Additionally, given these various memory arrangements, a POSITA would have been motivated to utilize the teachings of Saulsbury or Cullison, which each teach 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**
- **BreakingPoint Scalable System(s) + RFC 791**
- **BreakingPoint Scalable System(s) + Da Silva**
- **BreakingPoint Scalable System(s) + Cragon**
- **BreakingPoint Scalable System(s) + Ramanujan**
- **BreakingPoint Scalable System(s) + Dell PC System(s)**
- **BreakingPoint Scalable System(s) + IBM System(s)**
- **BreakingPoint Scalable System(s) + Intel Optane System(s)**
- **Any of the above + 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, such as the use of a chassis identifier, a module identifier, board identifier, instance identifier, and type identifier in packet headers. 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 their teachings—including details and features for chassis, and cards/blades, such as the use of a chassis identifier, module identifier, board identifier, instance identifier, and type identifier in packet headers—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.

A POSITA would have similarly been motivated to combine Scalable System(s) with RFC 791. BreakingPoint Scalable System(s) discloses a chassis-based architecture that sent packet communications throughout a network, which BreakingPoint Scalable System(s) teaches or suggests is an IP network over Ethernet. *See*, e.g., Exhibit 753-02. RFC 791 is the widely-known standard governing the IP protocol. Therefore, a POSITA would have known to use RFC 791 and been motivated to use the IP protocol with BreakingPoint Scalable System(s). For decades, the IP protocol has been ubiquitous in computers and computer networks- including for infrastructure testing computer systems such as BreakingPoint Scalable System(s). *See* RFC 791 at 1-3, 5-6, 8-9, and 11-23. Therefore, a POSITA would have looked to RFC 791 and been motivated to combine it with BreakingPoint Scalable System(s) packet-based network communications to facilitate packet-based communications over IP and Ethernet.

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 Ramanujan, which teaches a memory hierarchy and different configurable modes of memory operation. *See* Ramanujan at 3:32-35, 7:45-54, 8:11-29, 8:30-37, 9:1-10, 9:27-31, 9:34-40, 10:31-38, 13:3-9, 17:59-18:6, 31:5-12, 31:13-19, 31:29-32, FIG. 1, FIG. 2, FIG. 5B, FIG. 6D. Likewise, Cragon, teaches multiple memory arrangements and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491). A POSITA would have also been motivated to look at the memory arrangements of existing computer systems at the time, including Dell PC System(s), which teaches multiple memory arrangements, including for DDR3/DDR4; L1, L2, and L3 cache; flash memory; hard disks; etc. (*see* Exhibit 753-12) and IBM System(s), which teaches multiple memory arrangements, including DDR3; L1, L2, L3, and L4 cache; flash memory; hard disks; etc. for performing analysis of large volumes of data and operational analytics of workloads throughout a cloud network (*see* Exhibit 753-13). Additionally, given these various arrangements, a POSITA would have been motivated to utilize the teachings of Saulsbury 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**
- **SeaMicro Computing System(s) + RFC 791**
- **SeaMicro Computing System(s) + Da Silva**
- **SeaMicro Computing System(s) + Cragon**

- **SeaMicro Computing System(s) + Ramanujan**
- **SeaMicro Computing System(s) + Dell PC System(s)**
- **SeaMicro Computing System(s) + IBM System(s)**
- **SeaMicro Computing System(s) + Intel Optane System(s)**
- **Any of the above + 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," such as the use of a chassis identifier, module identifier, board identifier, instance identifier, and type identifier in packet headers. 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 their teachings—including details and features for chassis, and cards, such as the use of a chassis identifier, module identifier, board identifier,

instance identifier, and type identifier in packet headers—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.

A POSITA would have similarly been motivated to combine SeaMicro Computing System(s) with RFC 791. SeaMicro Computing System(s) discloses in a dense chassis configuration for networked, distributed computing, which SeaMicro Computing System(s) teaches or suggests is an IP network over Ethernet. *See,* e.g., Exhibit 753-03. RFC 791 is a widely-known standard governing the IP protocol. Therefore, a POSITA would have known to use RFC 791 and been motivated to use the IP protocol with SeaMicro Computing System(s). For decades, the IP protocol has been ubiquitous in computers and computer networks. *See* RFC 791 at 1-3, 5-6, 8-9, and 11-23. Therefore, a POSITA would have looked to RFC 791 and been motivated to combine it with SeaMicro Computing System(s) packet-based network communications to facilitate packet-based communications over IP and Ethernet.

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 SeaMicro Computing System(s) with Ramanujan, which teaches a memory hierarchy and different configurable modes of memory operation. *See* Ramanujan at 3:32-35, 7:45-54, 8:11-29, 8:30-37, 9:1-10, 9:27-31, 9:34-40, 10:31-38, 13:3-9, 17:59-18:6, 31:5-12, 31:13-19, 31:29-32, FIG. 1, FIG. 2, FIG. 5B, FIG. 6D. Likewise, Cragon teaches memory arrangements and modes of memory operation (*See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491). A POSITA would have also been motivated to look at the memory arrangements of existing computer systems at the time, including Dell PC System(s), which teaches multiple memory arrangements, including for DDR3/DDR4; L1, L3, and L3 cache; flash memory; hard disks; etc. (*See* Exhibit 753-12) and IBM System(s), which teaches multiple

memory arrangements, including DDR3; L1, L2, L3, and L4 cache; flash memory; hard disks; etc. for performing analysis of large volumes of data and operational analytics of workloads throughout a cloud network (*See* Exhibit 753-13). Additionally, given these various memory arrangements, a POSITA would have been motivated to utilize the teachings of Saulsbury 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**

- **Jana + RFC 791 + Ramanujan**
- **Jana + RFC 791 + Dell PC System(s)**
- **Jana + RFC 791 + IBM System(s)**
- **Jana + RFC 791 + Intel Optane System(s)**
- **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, Ramanujan, Dell PC System(s), IBM System(s), or Intel Optane System(s), which each teaches memory architectures and modes of memory operation (*see* charts 753-08, 753-12, 753-13, 753-14), and with Saulsbury 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**
- **Pierson + Stroud + Cragon**
- **Pierson + Stroud + Ramanujan**
- **Pierson + Stroud + Dell PC System(s)**
- **Pierson + Stroud + IBM System(s)**
- **Pierson + Stroud + Intel Optane System(s)**
- **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, Ramanujan, Dell PC System(s), IBM System(s), or Intel Optane System(s), which each teaches memory architectures and modes of memory operation (*see* charts 753-08, 753-12, 753-13, 753-14), and with Saulsbury 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.

- **IBM System(s)**
- **IBM System(s) + RFC 791**
- **IBM System(s) + Intel Optane System(s)**
- **IBM System(s) + Dell PC System(s)**
- **IBM System(s) + Ramanujan**
- **Any of the above + Cullison or Saulsbury**

A POSITA would have been motivated to combine IBM System(s) with RFC 791. The IBM System(s) discloses a computer server architecture, with teaching or suggestion that the network uses IP packets over Ethernet. RFC 791 is the widely-known standard governing the IP protocol. Therefore, a POSITA would have known to use RFC 791 and been motivated to use the IP protocol with IBM System(s). For decades, the IP protocol has been ubiquitous in computers and computer networks—including for servers and computers such as IBM System(s). *See* RFC 791 at 1-3, 5-6, 8-9, and 11-23. Therefore, a POSITA would have looked to RFC 791 and been motivated to combine it with IBM System(s) packet-based network communications to facilitate packet-based communications over IP and Ethernet.

Additionally, IBM System(s) support an assortment of memory functions and types, including DDR3; L1, L2, L3, and L4 cache; flash memory; hard disks; etc. *See, e.g.*, Exhibit 753-13. A POSITA would have looked to references that provide features for using different types of memory together. For example, a POSITA would have been motivated to combine IBM System(s) with Ramanujan, which teaches a memory hierarchy and different configurable modes of memory operation. *See* Ramanujan at 3:32-35, 7:45-54, 8:11-29, 8:30-37, 9:1-10, 9:27-31, 9:34-40, 10:31-38, 13:3-9, 17:59-18:6, 31:5-12, 31:13-19, 31:29-32, FIG. 1, FIG. 2, FIG. 5B, FIG. 6D. Likewise, Cragon teaches multiple memory arrangements and modes of memory operation. *See* Cragon at 73, 91, 110, 191, 231-232, 257, 414-415, 491. A POSITA would have also been motivated to look at the memory arrangements of existing computer systems at the time, including Dell PC System(s), which teaches multiple memory arrangements, including for DDR3/DDR4; L1, L2, and L3 cache; flash memory; hard disks; etc. *See* Exhibit 753-12. Additionally, given these various memory arrangements, a POSITA would have been motivated to utilize the teachings of Saulsbury or Cullison, which each teach 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.

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
Marcello Pogliani et al., *Quality of Service Driven Runtime Resource Allocation in Reconfigurable HPC Architectures*, in *Proceedings of the IEEE International Conference on Computational Science and Engineering (CSE)* (2016), https://ieeexplore.ieee.org/document/7982218.	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*, in *Proceedings of the Euromicro Conference on Digital System Design* (2013), https://ieeexplore.ieee.org/document/6628336.	O	Exhibit 943-07
Cray Supercomputer System(s), including but not limited to Cray XC40, Cray XD1, and Cray XE, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Cray Supercomputer System(s) were sold, offered for	A & O	Exhibit 943-08

Prior Art to ’943	Anticipation/ Obviousness	Claim Chart
sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Cray Supercomputer System(s) were publicly used by HPE/Cray, the public, and third-party customers before this date, by July 14, 2018 at the latest.		
AWS EC2 System, including at least F1 instances, was sold, offered for sale, made known, and/or publicly used before July 15, 2018. AWS EC2 System was sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. AWS EC2 System was publicly used by Amazon Web Services, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 943-09
MS Cloud Platform Solutions System(s), including at least Open CloudServers, Azure cloud servers, Catapult v2, v3, and Project BrainWave Storm Peak. were sold, offered for sale, made known, and/or publicly used before July 15, 2018. MS Cloud Platform Solutions System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. MS Cloud Platform Solutions System(s) were publicly used by Microsoft, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 943-10
UToronto SAVI System was sold, offered for sale, made known, and/or publicly used before July 15, 2018. UToronto SAVI System was sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. UToronto SAVI System was publicly used by the University of Toronto, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 943-11
IBM HPC System(s), including at least SuperVessel Cloud systems, Power8 CAPI systems, and cloudFPGA systems, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. IBM HPC System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this	A & O	Exhibit 943-12

Prior Art to ’943	Anticipation/ Obviousness	Claim Chart
date, by July 14, 2018 at the latest. IBM HPC System(s) ware publicly used by IBM, the public, and third-party customers before this date, by July 14, 2018 at the latest.		
Intel HARP/PAC System(s), including at least Intel HARP v1, Intel HARP v2, Intel PAC with Arria 10 GX FPGA, and MAP-80/100, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Intel HARP/PAC System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Intel HARP/PAC System(s) were publicly used by Intel/Altera, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 943-13
Xilinx Zynq Ultrascale+ System(s), including at least XCZU9EG, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Xilinx Zynq Ultrascale+ System(s) were sold, offered for sale, and/or made known to the public and third-party customers before this date, at least by Nov. 31, 2015. AMD-CV-0149659 (sales data for XCZU9EG). Xilinx Zynq Ultrascale+ System(s) were publicly used by Xilinx, the public, and third-party customers before this date, at least by Nov. 31, 2015.	A & O	Exhibit 943-14
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	
Naif Tarafdar et al., *Enabling Flexible Network FPGA Clusters in a Heterogeneous Cloud Data Center*, (Feb. 22–24, 2017),	O	

Prior Art to '943	Anticipation/ Obviousness	Claim Chart
https://tarafdar.github.io/files/papers/2017-02-22-enabling-flexible-network-fpga-clusters-in-a-heterogeneous-cloud-data-center.pdf.		
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**
- **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. This would have been the use of a known technique to improve similar devices in the same way.

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)**
- **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. This would have been applying a known technique to a known device ready for improvement to yield the predictable results of checking user permissions to increase security.

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)**
- **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. This would have been applying a known technique to a known device ready for improvement to yield the predictable results of checking user permissions to increase security.

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)**
- **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. This would have been a combination of prior art elements according to known methods to yield predictable results.

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).

- **Xilinx Zynq Ultrascale+ System(s)**
- **Xilinx Zynq Ultrascale+ System(s) + Cray Supercomputer System(s), AWS EC2 System, MS Cloud Platform Solutions System(s), IBM HPC System(s), or Bernat**
- **Any of the above + Jacobson or Schultz**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

The Xilinx Zynq Ultrascale+ System(s) discloses a use case of its SoCs in advanced data-center applications. *See* Ex. 943-14. Therefore, a POSITA would have been looked to combine the Xilinx Zynq Ultrascale+ System(s) with known configurations of data-centers, where servers typically included swappable pods or cards, for example as taught by Cray Supercomputer System(s), AWS EC2 System, MS Cloud Platform Solutions System(s), IBM HPC System(s), and Bernat. A POSITA would have been motivated to combine Xilinx Zynq Ultrascale+ System(s) into servers with swappable pods or cards because it is suggested by Xilinx Zynq Ultrascale+ System(s), and it was known in the art to be advantageous for servers to have modular, expandable designs, so that they can be manufactured at lower cost while allowing for greater flexibility. This would have been a combination of prior art elements according to known methods to yield predictable results.

The Xilinx Zynq Ultrascale+ System(s) discloses a system on chip ("SoC") that includes the programmable logic of an FPGA in one portion and a processing system ("PS") in another portion. *See* Ex. 943-14. The PS includes a configuration access port that allows software to configure the programmable logic portion. *See id.* Therefore, a POSITA would have been motivated to combine the Xilinx Zynq Ultrascale+ System(s) with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, Seshadri, or Atsatt-113. For example, 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. Moreover, Schultz is a patent from Xilinx disclosing techniques and configurations for use in PLDs, such as the Xilinx Zynq Ultrascale+ System(s). 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). A POSITA would naturally have looked to these references and been motivated to combine because it would have increased FPGA security when it is reprogrammed using software, which would have been beneficial for workstation, networked environments, and a variety of different use cases. Xilinx Zynq Ultrascale+ System(s) further suggests the combination to add such security features by disclosing that security was a major concern in the design and operation of such FPGA-based programmable logic devices. Ronaldo Architecture Specifications (AMD-CV-0137948–AMD-CV-0138306), §10 (discussing architecture goals and security measures); UG1228 Zynq UltraScale+ MPSoC Embedded Methodology Guide (AMD-CV-0119344–AMD-CV-0119560), §8 (discussing security needs and security methodology). This would have been a combination of prior art elements according to known methods to yield predictable results.

As an additional example, the Xilinx Zynq Ultrascale+ System(s) incorporated a PS to allow greater flexibility and use cases, including for applications that needed heterogenous processing using different types of hardware. Therefore, a POSITA would have been motivated to combine Xilinx Zynq Ultrascale+ System(s) 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 allowed different sections of the programmable logic to be used for different types of hardware acceleration, which furthers the goal of providing heterogenous computing

resources. This would have been the use of a known technique to improve similar devices in the same way.

As an additional example, the Xilinx Zynq Ultrascale+ System(s) discloses a high-performance SoC for data processing in data-centers. Given the constraints on power, thermal budgets, and network bandwidth in the relevant timeframe—which had been widely recognized as major constraints on computer performance by that time—a POSITA would have looked to combine the Xilinx Zynq Ultrascale+ 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 Xilinx Zynq Ultrascale+ System(s) with such references because it would have improved the power and thermal constraints on performance of the FPGA and increased efficiency of FPGA resources, 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 Xilinx Zynq Ultrascale+ 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 Xilinx Zynq Ultrascale+ 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 Xilinx Zynq Ultrascale+ System(s) includes an SoC/FPGA with various I/O ports for communications. Therefore, a POSITA would have been motivated to combine the Xilinx Zynq Ultrascale+ 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 Xilinx Zynq Ultrascale+ 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 Xilinx Zynq Ultrascale+ System(s). The Petition in IPR2025-00223 explained the motivation to combine Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations. The Xilinx Zynq Ultrascale+ System(s) was an FPGA-based SoC. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the Xilinx Zynq Ultrascale+ System(s), and POSITA would have been motivated to combine for similar reasons.

- **Intel HARP/PAC System(s)**
- **Intel HARP/PAC System(s) + Cray Supercomputer System(s), AWS EC2 System, MS Cloud Platform Solutions System(s), IBM HPC System(s), or Bernat**
- **Any of the above + Jacobson or Schultz**
- **Any of the above + Cervero**
- **Any of the above + Pogliani or Bernat**
- **Any of the above + Abdelfattah**
- **Any of the above + Rahim**

The Intel HARP/PAC System(s) discloses accelerator cards/boards that were often used in data-center applications. Therefore, a POSITA would have been looked to combine the Intel HARP/PAC System(s) with known configurations of data-centers, where servers typically included swappable pods or cards, for example as taught by Cray Supercomputer System(s), AWS EC2 System, MS Cloud Platform Solutions System(s), IBM HPC System(s), and Bernat. A POSITA would have been motivated to combine Intel HARP/PAC System(s) into servers with swappable pods or cards because it is suggested by Intel HARP/PAC System(s), and it was known in the art to be advantageous for servers to have modular, expandable designs, so that they can be

manufactured at lower cost while allowing for greater flexibility. This would have been a combination of prior art elements according to known methods to yield predictable results.

The Intel HARP/PAC System(s) discloses accelerator card or boards that include an FPGA with multiple Accelerator Functional Units (AFU) loaded to Partial Reconfiguration (PR) partitions, encapsulated by an FPGA Interface Manager (FIM). *See* Ex. 943-13. The FIM includes a port and bridge that controls access to configure the PR partitions, e.g., by loading different AFUs. *See id*. Therefore, a POSITA would have been motivated to combine the Intel HARP/PAC System(s) with known techniques for locking an FPGA against unauthorized access, including but not limited to Jacobson, Schultz, Seshadri, or Atsatt-113. For example, 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 in FPGAs. *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, Intel HARP/PAC System(s) are Intel products that used Intel/Altera FPGAS 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 when it is reprogrammed using software, which would have been beneficial for workstation, networked environments, and a variety of different use cases. This would have been a combination of prior art elements according to known methods to yield predictable results.

As an additional example, the Intel HARP/PAC System(s) discloses the flexible assignment of multiple PR partitions, in multi-user environments, with users loading their own custom logic AFUs. *See* Ex. 943-13. Therefore, a POSITA would have been motivated to combine Intel HARP/PAC System(s) 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. Additionally, the Intel HARP/PAC System(s) discloses heterogenous processing platforms with FPGAs and Xeon processors. The combination with Cervero would have allowed different sections of the programmable logic to be used for different types of hardware acceleration, which furthers the goal of providing heterogenous computing resources. This would have been the use of a known technique to improve similar devices in the same way.

As an additional example, the Intel HARP/PAC System(s) discloses a card for data processing in data-centers. Given the constraints on power, thermal budgets, and network bandwidth in the relevant timeframe—which had been widely recognized as major constraints on computer performance by that time—a POSITA would have looked to combine the Intel HARP/PAC 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. Additionally, Bernat is a patent application from Intel describing similar data processing platforms to the Intel HARP/PAC System(s). A POSITA would have looked to and been motivated to combine the Intel HARP/PAC System(s) with such references because it would have improved the power and thermal constraints on performance of the FPGA and increased efficiency of FPGA resources, 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 Intel HARP/PAC 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, Rahim is an Intel/Altera patent describing techniques developed for application to Intel/Altera

FPGAs. A POSITA would have looked to and been motivated to combine the Intel HARP/PAC 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 Intel HARP/PAC System(s) includes an SoC/FPGA with various I/O ports for communications. Therefore, a POSITA would have been motivated to combine the Intel HARP/PAC 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 Intel HARP/PAC 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 Intel HARP/PAC System(s). The Petition in IPR2025-00223 explained the motivation to combine Biederman, Tarafdar, Kulkarni, and/or Atsatt-923 with known PLD/FPGA configurations. The Intel HARP/PAC System(s) was an FPGA-based SoC. Therefore, the motivations to combine explained in the incorporated Petition (IPR2025-00223) also apply to the Intel HARP/PAC System(s), and POSITA would have been motivated to combine for similar reasons.

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”), including but not limited to Verizon switches, routers, gateways, and access points, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Verizon Networking Devices were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Verizon Networking Devices were publicly used by Verizon, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 634-01
Verizon Communications Inc.’s networking solutions (“Verizon Networking Solutions”), including but not limited to Verizon Managed SD WAN, Verizon Managed LAN Services, Verizon Managed SD Branch, and Verizon Secure Gateway Services, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Verizon Networking Solutions were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Verizon Networking Solutions were publicly used by Verizon, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 634-02
Verizon Communications Inc.’s Voice over Internet Protocol devices (“Verizon VoIP Products”), including but not limited to Fios Digital Voice, Verizon One Talk, VoIP Business Digital Voice, and Preferred Voice, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Verizon VoIP Products were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Verizon VoIP Products were publicly used by Verizon, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 634-03

Prior Art to ’634	Anticipation/ Obviousness	Claim Chart
Google LLC’s networking products (“Google Networking Products”), including but not limited to Google Fi, Google OnHub, Google Wifi, Google Voice, Google Meet, and Google Hangout, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. Google Networking Products were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. Google Networking Products were publicly used by Google, the public, and third-party customers before this date, by July 14, 2018 at the latest.	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
Systems implementing the Congestion Manager developed by a group of researchers at Massachusetts Institute of Technology, Carnegie Mellon University, and other institutions (“Congestion Manager System”) were sold, offered for sale, made known, and/or publicly used before July 15, 2018. The Congestion Manager System were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. The Congestion Manager System were publicly used by its developers, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 634-11

Prior Art to ’634	Anticipation/ Obviousness	Claim Chart
David G. Andersen et al., *System Support for Bandwidth Management and Content Adaptation in Internet Applications*, in Proceedings of the 4th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’00) (USENIX Ass’n 2000), https://www.usenix.org/legacy/publications/library/proceedings/osdi2000/full_papers/andersonbansal/andersonbansal.pdf.	A & O	Exhibit 634-12
RealNetworks’ system and platform for delivering and streaming media content (“RealNetworks Media Delivery System”), including but not limited to Real Broadcast Network, RealSystem, RealServer, RealPlayer, Helix Server, and RealProducer, were sold, offered for sale, made known, and/or publicly used before July 15, 2018. RealNetworks Media Delivery System were sold, offered for sale, and/or made known to the public and third-party customers before this date, by July 14, 2018 at the latest. RealNetworks Media Delivery System were publicly used by RealNetworks, the public, and third-party customers before this date, by July 14, 2018 at the latest.	A & O	Exhibit 634-13

In addition to the anticipation and obviousness explained in this section, the ’634 patent is anticipated and/or rendered obvious by prior art as explained by Defendants’ Petition for Inter Partes Review (assigned Case No. IPR2025-00479), which is incorporated herein by reference:

Grounds	Claims	Prior Art
1	1-20	Erskine
2	1-20	Scherzer
3	1-20	Scherzer in view of Gillett

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 + Congestion Manager System**

- **Verizon Networking Devices + Congestion Manager System + Sella**
- **Verizon Networking Devices + Congestion Manager System + Ullmann**
- **Verizon Networking Devices + Congestion Manager System + Erskine**
- **Verizon Networking Devices + Congestion Manager System + Blaisdell**
- **Verizon Networking Devices + Congestion Manager System + Scherzer**
- **Verizon Networking Devices + Congestion Manager System + Gillett**
- **Verizon Networking Devices + Andersen**
- **Verizon Networking Devices + Andersen + Sella**
- **Verizon Networking Devices + Andersen + Ullmann**
- **Verizon Networking Devices + Andersen + Erskine**
- **Verizon Networking Devices + Andersen + Blaisdell**
- **Verizon Networking Devices + Andersen + Scherzer**
- **Verizon Networking Devices + Andersen + 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.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Congestion Manager System, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Devices with Andersen, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

- **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 + Congestion Manager System**
- **Verizon Networking Solutions + Congestion Manager System + Sella**
- **Verizon Networking Solutions + Congestion Manager System + Ullmann**
- **Verizon Networking Solutions + Congestion Manager System + Erskine**
- **Verizon Networking Solutions + Congestion Manager System + Blaisdell**
- **Verizon Networking Solutions + Congestion Manager System + Scherzer**
- **Verizon Networking Solutions + Congestion Manager System + Gillett**
- **Verizon Networking Solutions + Andersen**
- **Verizon Networking Solutions + Andersen + Sella**
- **Verizon Networking Solutions + Andersen + Ullmann**
- **Verizon Networking Solutions + Andersen + Erskine**
- **Verizon Networking Solutions + Andersen + Blaisdell**
- **Verizon Networking Solutions + Andersen + Scherzer**
- **Verizon Networking Solutions + Andersen + 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.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Congestion Manager System, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

As an additional example, a POSITA would have been motivated to combine Verizon Networking Solutions with Andersen, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

- **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 + Congestion Manager System**
- **Verizon VoIP Products + Congestion Manager System + Sella**
- **Verizon VoIP Products + Congestion Manager System + Ullmann**
- **Verizon VoIP Products + Congestion Manager System + Erskine**
- **Verizon VoIP Products + Congestion Manager System + Blaisdell**

- **Verizon VoIP Products + Congestion Manager System + Scherzer**
- **Verizon VoIP Products + Congestion Manager System + Gillett**
- **Verizon VoIP Products + Andersen**
- **Verizon VoIP Products + Andersen + Sella**
- **Verizon VoIP Products + Andersen + Ullmann**
- **Verizon VoIP Products + Andersen + Erskine**
- **Verizon VoIP Products + Andersen + Blaisdell**
- **Verizon VoIP Products + Andersen + Scherzer**
- **Verizon VoIP Products + Andersen + 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.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Congestion Manager System, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

As an additional example, a POSITA would have been motivated to combine Verizon VoIP Products with Andersen, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

- **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 + Congestion Manager System**
- **Google Networking Products + Congestion Manager System + Sella**
- **Google Networking Products + Congestion Manager System + Ullmann**
- **Google Networking Products + Congestion Manager System + Erskine**
- **Google Networking Products + Congestion Manager System + Blaisdell**
- **Google Networking Products + Congestion Manager System + Scherzer**
- **Google Networking Products + Congestion Manager System + Gillett**
- **Google Networking Products + Andersen**
- **Google Networking Products + Andersen + Sella**
- **Google Networking Products + Andersen + Ullmann**
- **Google Networking Products + Andersen + Erskine**
- **Google Networking Products + Andersen + Blaisdell**
- **Google Networking Products + Andersen + Scherzer**
- **Google Networking Products + Andersen + 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.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Congestion Manager System, which teaches collecting network

characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

As an additional example, a POSITA would have been motivated to combine Google Networking Products with Andersen, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

- **Congestion Manager System + Sella**
- **Congestion Manager System + Sella + Ullmann**
- **Congestion Manager System + Sella + Erskine**
- **Congestion Manager System + Blaisdell**
- **Congestion Manager System + Blaisdell + Scherzer**
- **Congestion Manager System + Blaisdell + Gillett**
- **Congestion Manager System + Erskine**
- **Congestion Manager System + Erskine + Ullmann**
- **Congestion Manager System + Scherzer**
- **Congestion Manager System + Scherzer + Gillett**

The Congestion Manager System is described in multiple documents including those listed in Exhibit 634-11. One of ordinary skill in the art would have understood that these documents describe the Congestion Manager System and would have been motivated to combine the listed documents because they all describe the Congestion Manager System, share authors, 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, the Congestion Manager System disclose devices, methods, systems, and architectures for establishing and managing networking connections using various network parameters to optimize network performance. For example, the Congestion Manager System includes a Congestion Manager ("CM"), which "ensures proper congestion behavior and allows applications to easily adapt to network

congestion," "maintains congestion parameters and exposes an API to enable applications to learn about network characteristics, pass information to the CM, and schedule data transmissions," and "uses a window-based control algorithm, a scheduler to regulate transmissions, and a lightweight protocol to elicit feedback from receivers." *See, e.g.*, Exhibit 634-11, An Integrated Congestion Management Architecture for Internet Hosts. 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 Congestion Manager System with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine Congestion Manager System with Ullmann, which teaches monitoring and managing network bandwidth.

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

- **Andersen + Sella**
- **Andersen + Sella + Ullmann**
- **Andersen + Sella + Erskine**
- **Andersen + Blaisdell**

- **Andersen + Blaisdell + Scherzer**
- **Andersen + Blaisdell + Gillett**
- **Andersen + Erskine**
- **Andersen + Erskine + Ullmann**
- **Andersen + Scherzer**
- **Andersen + Scherzer + Gillett**

Andersen discloses devices, methods, systems, and architectures for establishing and managing networking connections using various network parameters to optimize network performance. For example, Andersen teaches that a Congestion Manager ("CM") "performs two important functions": "First, it enables efficient multiplexing and congestion control by integrating congestion management across multiple flows. Second, it enables efficient application adaptation to congestion by exposing its knowledge of network conditions to applications." *See, e.g.*, Exhibit 634-12 (Andersen, 3). The CM "assigns [a] flow to the appropriate macroflow based on its destination" and permits a client to transmit data based on available rate. *Id*. The CM also provides an API that "specifies functions for managing state, for performing data transmissions, for applications to inform the CM of losses, for querying the CM about network state, and for constructing and splitting macroflows if the default per-destination aggregation is unsuitable for an application." *See, e.g.*, Exhibit 634-12 (Andersen, 4). 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 Andersen with Sella, which teaches dynamically changing bandwidth allocation.

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

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

- **RealNetworks Media Delivery System + Sella**
- **RealNetworks Media Delivery System + Sella + Ullmann**
- **RealNetworks Media Delivery System + Sella + Erskine**
- **RealNetworks Media Delivery System + Blaisdell**
- **RealNetworks Media Delivery System + Blaisdell + Scherzer**
- **RealNetworks Media Delivery System + Blaisdell + Gillett**
- **RealNetworks Media Delivery System + Erskine**
- **RealNetworks Media Delivery System + Erskine + Ullmann**
- **RealNetworks Media Delivery System + Scherzer**
- **RealNetworks Media Delivery System + Scherzer + Gillett**
- **RealNetworks Media Delivery System + Congestion Manager System**
- **RealNetworks Media Delivery System + Congestion Manager System + Sella**
- **RealNetworks Media Delivery System + Congestion Manager System + Ullmann**

- **RealNetworks Media Delivery System + Congestion Manager System + Erskine**
- **RealNetworks Media Delivery System + Congestion Manager System + Blaisdell**
- **RealNetworks Media Delivery System + Congestion Manager System + Scherzer**
- **RealNetworks Media Delivery System + Congestion Manager System + Gillett**
- **RealNetworks Media Delivery System + Andersen**
- **RealNetworks Media Delivery System + Andersen + Sella**
- **RealNetworks Media Delivery System + Andersen + Ullmann**
- **RealNetworks Media Delivery System + Andersen + Erskine**
- **RealNetworks Media Delivery System + Andersen + Blaisdell**
- **RealNetworks Media Delivery System + Andersen + Scherzer**
- **RealNetworks Media Delivery System + Andersen + Gillett**

RealNetworks Media Delivery System is described in numerous documents including those listed in Exhibit 634-13. One of ordinary skill in the art would have understood that these documents describe RealNetworks' system and platform for delivering and streaming media content and would have been motivated to combine the listed documents because they all describe RealNetworks' media delivery system, 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, RealNetworks Media Delivery System discloses devices, methods, systems, and architectures for establishing and managing network connections based on various network parameters, optimizing network and media delivery performance, and initiating and controlling transmission of data over the network. For example, RealNetworks Media Delivery System includes intelligent routing software that routes users "to the topographically closest or least busy splitter" where content can be accessed. *See, e.g.*, Exhibit 634-13, Introducing the First Distributed Multi-Tier Broadcast Architecture for the Internet. RealNetworks Media Delivery System also provides capabilities such as gathering

statistics about usage and quality of service, capacity sharing, fail over to alternative servers, and adaptively changing the bandwidth of streams. *See, e.g.*, Exhibit 634-13. 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 RealNetworks Media Delivery System with Sella, which teaches dynamically changing bandwidth allocation.

As an additional example, a POSITA would have been motivated to combine RealNetworks Media Delivery System with Ullmann, which teaches monitoring and managing network bandwidth.

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

As an additional example, a POSITA would have been motivated to combine RealNetworks Media Delivery System with Congestion Manager System, which teaches collecting network characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

As an additional example, a POSITA would have been motivated to combine RealNetworks Media Delivery System with Andersen, which teaches collecting network

characteristics, coordinating host- and domain-specific path information, sharing path properties between different streams, and apportioning bandwidths to flows.

II. Identification of Limitations that Defendants Contend Lack Written Description, Lack Enablement, or are Indefinite Under 35 U.S.C. § 112

The Asserted Claims recite limitations that were already well known in the prior art (and therefore are anticipated and obvious). If the claims are construed to recite some novel limitation or combination of limitations, however, then they fail to describe or enable that limitation or combination, including how it differs from what was already known.

Section 112 "contains a written description requirement separate from enablement." *Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010). The written description requirement requires that "the disclosure of the application relied upon reasonably convey[] to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Id.* Indeed, "the specification must describe an invention understandable to [a] skilled artisan and show that the inventor actually invented the invention claimed." *Id.*

In addition to the written description requirement, Section 112 imposes an enablement requirement, which is satisfied only when "one skilled in the art, after reading the specification, could practice the claimed invention without undue experimentation." *Sitrick v. Dreamworks, LLC*, 516 F.3d 993, 999 (Fed. Cir. 2008). Further, "[t]he full scope of the claimed invention must be enabled," and "[a] patentee who chooses broad claim language must make sure the broad claims are fully enabled." *Id.*

Many of the Asserted Claims are so poorly drafted as to be indecipherable to anyone, including those skilled in the art, and are therefore facially indefinite and invalid. To be valid, a patent claim must "particularly point [] out and distinctly claim[] the subject matter which the applicant . . . regards as his invention." 35 U.S.C. § 112(b). The purpose of this definiteness requirement is "afford clear notice of what is claimed, thereby apprising the public of what is still open to them." *Nautilus v. Biosig Instruments, Inc.*, 572 U.S. 898, 909 (2014) (internal quotations omitted). Patent claims are invalid for indefiniteness when they fail to "inform those skilled in the art about the scope of the invention with reasonable certainty." *Id.* at 910.

Many of the Asserted Claims fail to satisfy the written-description, enablement, or definiteness requirements of § 112 as explained herein.

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

For the claim terms "queue implemented in hardware" and "register" in the '596 patent, Defendants' claim construction briefing explained how Plaintiffs' claim interpretations and proposed constructions rendered these limitations indefinite under 35 U.S.C. §112; that analysis and explanation is hereby incorporated by reference. Dkts. 79, 81.

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. The basis for this contention is disclosed in Defendants' claim construction briefing for the '596 patent (Dkts. 79, 81), and is hereby incorporated by reference.

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," lacks enablement, and fails to clearly define the term, rendering it indefinite.

For example, 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 demonstrate that the inventors were in possession of the specific structure of such a protocol at the time of filing, as it 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 offers the circular explanation "[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.

One of ordinary skill in the art at the time of the invention would have expected the inventors to describe major aspects of a novel storage protocol's design if they were in possession of the protocol at the time of the claimed invention. The following examples are just some aspects of a storage protocol that a POSITA would have expected to see described if the inventors possessed such a protocol:

- Grammar of the protocol: The specification of the commands that are a part of the protocol. This requires consideration of the native capabilities of the storage devices supported by the protocol (e.g., the native command set). This also

includes the properties of individual commands included in the grammar (e.g., atomicity, consistency, idempotency, reliability, availability, and durability).

- Semantics of the protocol: The meaning of the commands.
- Data Structures: The specification of data types and structures, such as queues, and handling of data structures.
- Message format and syntax: The specification of the content of messages, formatting of messages, ordering of messages, etc.
- Error handling: The specification of error conditions, their properties (e.g., hard and soft faults), how they are represented, error correction, and how handled.
- Allocation of responsibilities between the components of a system (e.g., the host, the storage media device, and their associated interfaces and controllers).

The patent's specification does not adequately describe an abstraction protocol, nor does the patent's specification identify any then-existing protocol that constitutes the claimed abstraction protocol. However, Plaintiffs' infringement contentions identify NVME over Fabrics as the alleged abstraction protocol in the Accused Instrumentalities and states in part:

> NVMe over Fabrics is an abstraction protocol, and relies on the underlying NVMe Transport to ensure reliable delivery of NVMe commands and data. NVMe over Fabrics also relies on RDMA. Defendants' implementation and use of NVMe over Fabrics by itself and as implementing and using NVMe Transport and RDMA is a claimed abstraction protocol. The underlying NVMe Transport (as well as RDMA) serves as a part of the NVME over Fabrics abstraction protocol, independent of any specific physical interconnect characteristics. NVMe over Fabrics is a full-duplex protocol and supports full command queuing for the at least one storage media device. Further, NVMe over Fabrics is performance-centric, supports common high-level read and write access to a pool of storage media devices, is link-agnostic; and may be carried via a multiplicity of direct attach or networked interfaces. NVMe-oF corresponds to the claimed abstraction protocol in that it provides a performance-centric storage abstraction layer that exposes uniform high-level read and write operations to a pool of storage media devices, independent of the physical characteristics of the underlying media. NVMe-oF abstracts the storage devices from the host and maintains efficient command queuing and parallel execution semantics. NVMe-oF is link-agnostic, as it supports carriage over multiple transport mechanisms, including RDMA-based fabrics and TCP/IP networks, and is deployable over both direct-attach and networked interfaces. Further, NVMe-oF permits heterogeneous storage implementations behind the abstraction layer while presenting a consistent protocol interface to the host.

12/30/25 Inf. Contentions, Ex. B at 16-17 (citations omitted).

NVME is a storage protocol that was standardized by the NVMExpress organization to provide access to storage devices over PCIe links. NVME over Fabric is another specification standardized by the NVMExpress organization to provide encapsulation and transport of NVME commands over fabrics instead of PCIe links. Each of the specifications are hundreds of pages long and reflect over a hundred thousand hours of time spent by engineers to develop working protocols. These standards themselves are further evidence of why a person of ordinary skill would not have understood the applicant to be in possession of the claimed subject matter because they illustrate aspects of a storage protocol that the applicants would have disclosed in the specification had they been in possession of an abstraction protocol.

For example, the NVME specification illustrates the following aspects of a protocol that the '767 patent does not disclose:

- Data structures
- Command sets
- Control architecture
- Error reporting and recovery

For example, the NVME over Fabrics specification illustrates the following aspects of a protocol that the '767 patent does not disclose:

- Encapsulation and data transfers
- Commands
- Control architecture
- Discovery
- Authentication
- Transport (e.g., queueing, error recovery, etc.)

Moreover, the '767 patent specification does not enable one of ordinary skill in the art to practice the full scope of the claimed invention without undue experimentation for a number of reasons. As described above, there are a number of major aspects that must be specified for a storage protocol and the patent specification does not describe any of them. In addition, the NVME and NVME over Fabric standards, which Plaintiffs contend are examples of an abstraction

protocol, required over one hundred thousand hours of development effort to produce working storage protocols. The patent specification does not provide any guidance to a person ordinary on how to make any of these design decisions, let alone make design choices that achieve the various properties the patent specification states the abstraction protocol must have.

"Abstraction protocol" is also indefinite. 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. The basis for this contention is further disclosed in Defendants' claim construction briefing for the '767 patent (Dkts. 79, 81), and is hereby incorporated by reference.

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

For the claim terms ""interleaved memory" and "swapped memory over a specified address range" in the '753 patent, Defendants' claim construction briefing explained how Plaintiffs' claim interpretations and proposed constructions rendered these limitations indefinite under 35 U.S.C. §112; that analysis and explanation is hereby incorporated by reference. Dkts. 79, 81.

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" and "user-defined functionality" on a PLD '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). The basis for this contention is further disclosed in Defendants' claim construction briefing for the '943 patent (Dkt. 79); that analysis is hereby incorporated by reference.

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. The basis for this contention is further disclosed in Defendants' claim construction briefing for the '753 patent (Dkts. 79, 81); that analysis is hereby incorporated by reference.

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

For the claim term "programmable logic device" in the '943 patent, Defendants' claim construction briefing explained how Plaintiffs' claim interpretations and proposed constructions rendered these limitations indefinite under 35 U.S.C. §112; that analysis and explanation is hereby incorporated by reference. Dkts. 79, 81.

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). The basis for this contention is further disclosed in Defendants' claim construction briefing for the '943 patent (Dkt. 79), and is hereby incorporated by reference.

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

1. "network connected device"

Claims 15-20 recite "network connected device" and are invalid as indefinite under 35 U.S.C. § 112. The term "network connected device," as used in claims 15-20, 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.

Claim 15 recites that a network connected device is "acting as a tuner server." '634 patent, 15:46-47. Claims 16 and 20 recite the "network connected device of claim 15." '634 patent, 15:9, 16:37. At least claims 15, 16, and 20 include limitations that require the network connected device to function as a distinct device or to function circularly. For example, Claim 15 requires the "network connected device" to "collect; at a first endpoint device coupled to the network; parameter values for determination of bandwidth of network connections to further endpoint devices," where "the network connected device acting as a tuner server [is] distinct from the first endpoint device." '634 patent, 15:42-48. This requires the network connected device that acts as a distinct device to perform the collecting action. As another example, claim 16 requires the "network connected device" to "send the parameter values for determination of bandwidth, from

the first endpoint device, to the tuner server," i.e., to send parameter values to itself, which is circular. *Id.*, 16:11. The claim also requires that the network connected device "send, from the first endpoint device to the tuner server, connection information for the next network connection" and "receive, at the first endpoint device, from the tuner server, the one or more parameter values for the transmission bandwidth," which is also circular. *Id.*, 16:15, 20. Similarly, claim 20 requires the "network connected device" to "transmit data associated with the network traffic from the first endpoint to the network connected device" and to "receive data for adjusting the transmission bandwidth from the network connected device," i.e., transmitting data to and receiving data from itself, which is likewise circular. *Id.*, 16:41-42, 45-46. Because claim 15 and claims 16-20 that depend on claim 15 require a network connected device to function as a distinct device or to function circularly, a POSITA would not have understood their meaning. Thus, claims 15-20 are invalid under § 112. The basis for this contention is disclosed in Defendants' claim construction briefing for the '634 patent (Dkts. 79, 81), and is hereby incorporated by reference.

Moreover, claims 15-20 are invalid for lack of written description. For example, the specification of the '634 patent fails to describe how one or more processors at a tuner server can collect parameter values, receive connection information, or receive data at an endpoint device that is distinct from the tuner server. A POSITA reading the '634 patent would not have understood that its inventors had possession of processors at the tuner server performing the aforementioned functionalities at the endpoint device. In their claim construction brief (Dkt. 80), Plaintiffs argued that the network connected device "control[s]" the endpoint device to perform operations. This argument contradicts the claim language and essentially requires redrafting the claims. But even if one were to accept Plaintiffs' argument, written description would still be lacking. In particular, the specification lacks written description of control of the tuner server over the endpoint device. For example, a POSITA would have expected that any control of the tuner server over the endpoint device would require a mechanism for the tuner server to exert such control. However, the specification does not describe the tuner server providing instructions or commands to endpoint devices to perform the aforementioned operations, nor does the

specification describe communication protocols or mechanisms for the tuner server to control the endpoint devices.

Claims 15-20 are also invalid for lack of enablement. The ’634 patent does not include any description that would have allowed a POSITA to implement processors at a network connected device acting as a tuner server that collect parameter values, receive connection information, or receive data at an endpoint device that is distinct from the tuner server. A POSITA would not have known how such a configuration can be implemented and would not have been able to implement the claimed invention in claims 15-20 without undue experimentation, if at all.

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

Claims 1, 8, and 15, and all claims depending from them are invalid under 35 U.S.C. § 112 because the ’634 patent fails to provide sufficient written description of the claimed determination that a next network connection matches a geographical area of a past network connection. The ’634 patent describe examples of geographical area to include “an apartment complex, neighborhood, or city to city or state to state network connection” ’634 patent, 2:54-56. For example, a POSITA would have understood that there was no commonly known way to determine the apartment complex that a network connection is in or to determine whether two different connections are in the same apartment complex. The ’634 patent includes no description as to how to determine two network connections have matching geographical areas. Given its lack of relevant description, a POSITA reading the ’634 patent would have understood that the inventors did not have possession of this claimed feature.

All claims of the ’634 patent are also invalid for lack of enablement. The ’634 patent does not include any description that would have informed a POSITA as to how to determine that two network connections match in terms of geographical area, such as being in the same neighborhood. A POSITA would not have understood any such determination to be practical or reliable. At the very least, because the ’634 patent does not disclose any method of determining that a network connection matches the geographical area of another network connection, a

POSITA would not have been able to implement the claimed invention without undue experimentation.

3. "initiate/initiating the next network connection … with a transmission bandwidth based on the parameter values for the past network connection"

Claims 1, 8, and 15 and all claims depending from them are invalid under 35 U.S.C. § 112 because the '634 patent fails to provide sufficient written description of initiating the next network connection with a transmission bandwidth based on the parameter values for the past network connection. For example, a POSITA would have understood that commonly known and widely used transport layer protocols, such as TCP, use dynamic and adaptively changing transmission bandwidth. The transmission bandwidth of a network connection often depends upon factors that are not fully controllable based on settings and parameters. The '634 patent does not describe how to initiate a next network connection with a transmission bandwidth based on parameter values of a past network connection. A POSITA reading the '634 patent would have understood that the inventors did not have possession of this claimed feature.

All claims of the '634 patent are also invalid for lack of enablement. The '634 patent does not include any description that would have informed a POSITA as to how to initiate a new network connection with a transmission bandwidth based on parameter values of a past network connection. A POSITA would have found various difficulties in attempting to implement this feature under common schemes such as TCP, as this feature deviates from TCP's mechanism of dynamically adjusting the transmission bandwidth using a mechanism other than one that is based on parameter values of a past connections. A POSITA would not have been able to implement this claimed feature without undue experimentation.

4. "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. Claim 17 depends from claim 15, which recites that "the network connected device [is] distinct from the first endpoint device." The '634 patent, however, lacks written description for the claimed feature in

claim 17. For example, the '634 patent does not describe the structure of the network connected device such that it is both distinct from the first endpoint device and contains the first endpoint device. Furthermore, the '634 patent describes "offload[ing] the storage and bandwidth determination (e.g., calculations) from the endpoints … to the tuning server" ('634 patent, 4:42:44) but does not describe how that can be achieved when the network connected device, acting as the tuner server, contains the first endpoint device. A POSITA reading the '634 patent would have understood that the inventors did not have possession of claimed feature in claim 17.

Claim 17 is also invalid for lack of enablement. The '634 patent does not include any description that would have informed a POSITA as to how to structure the network connected device and the first endpoint device such that the former is distinct from, but contains, the latter. In light of the absence of description, a POSA would not have been able to implement this claimed feature without undue experimentation.

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. 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 '767 patent (Dkt. 156), which is hereby incorporated by reference, and further disclosed herein.

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’ motions to dismiss for failure to state a claim under 35 U.S.C. § 101 for the ’634 patent (Dkt. 23, 32, 156), 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. Document Production Accompanying Invalidity Contentions

Patent L.R. 3-4(a) ("Source code, specifications, schematics, flow charts, artwork, formulas, or other documentation sufficient to show the operation of any aspects or elements of an Accused Instrumentality identified by the patent claimant in its Patent L.R. 3-1(c) chart."): Defendants have produced the source code for the Accused Instrumentalities for inspection at the office of Sheppard Mullin in Menlo Park, California.

Patent L.R. 3-4(b) ("A copy or sample of the prior art identified pursuant to Patent L.R. 3-3(a) which does not appear in the file history of the patent(s) at issue. To the extent any such item is not in English, an English translation of the portion(s) relied upon shall be produced."): Defendants have produced the prior art referenced in these invalidity contentions to Plaintiffs, including in the documents bates labeled PA000001–PA040826.

Additional prior art referenced in these invalidity contentions is being produced to Plaintiffs concurrently with service of these invalidity contentions labeled PA040827–PA068833, AMD-CV-0065505–AMD-CV-0065651, AMD-CV-0110842–AMD-CV-0149658. Source code for prior art has been made available for inspection.

Patent L.R. 3-4(c) ("All agreements that may be related to the accused instrumentality or may be comparable to a license that would result from a hypothetical reasonable royalty negotiation."):

Defendants have produced responsive agreements to Plaintiffs, including documents bates labeled: AMD-CV-0024567; AMD-CV-0030880; AMD-CV-0096930; AMD-CV-0096945; AMD-CV-0097020; AMD-CV-0097025; AMD-CV-0097084; AMD-CV-0109074; AMD-CV-0109099; AMD-CV-0109127; AMD-CV-0109152; AMD-CV-0109182; AMD-CV-0109334; AMD-CV-0109351; AMD-CV-0109381; AMD-CV-0109392; AMD-CV-0109404; AMD-CV-0109417; AMD-CV-0109837; AMD-CV-0109902; AMD-CV-0109906; AMD-CV-0110685; AMD-CV-0110688; AMD-CV-0110693; AMD-CV-0110761; AMD-CV-0110786; AMD-CV-0110802; AMD-CV-0110817.

Additional responsive agreements are being produced to Plaintiffs.

Patent L.R. 3-4(d) ("Documents sufficient to show the sales, revenue, cost, and profits for accused instrumentalities identified pursuant to Patent L.R. 3-1(b) for any period of alleged infringement."):

Defendants have produced responsive documents showing sales, revenue, cost, and profits for accused instrumentalities to Plaintiffs, including documents bates labeled: AMD-CV-0005255; AMD-CV-0005256; AMD-CV-0008996; AMD-CV-0063058; AMD-CV-0063196; AMD-CV-0063299; AMD-CV-0063375; AMD-CV-0063457; AMD-CV-0065662; AMD-CV-0065783; AMD-CV-0065958; AMD-CV-0066066; AMD-CV-0066128; AMD-CV-0066252; AMD-CV-0066341; AMD-CV-0066419; AMD-CV-0109441; AMD-CV-0109523; AMD-CV-0109626; AMD-CV-0109698; AMD-CV-0109836.

Patent L.R. 3-4(e) ("All agreements that may be used to support the party denying infringement's damages case."):

Defendants have produced responsive agreements to Plaintiffs, including documents bates labeled: AMD-CV-0024567; AMD-CV-0030880; AMD-CV-0096930; AMD-CV-0096945; AMD-CV-0097020; AMD-CV-0097025; AMD-CV-0097084; AMD-CV-0109074; AMD-CV-0109099; AMD-CV-0109127; AMD-CV-0109152; AMD-CV-0109182; AMD-CV-0109334; AMD-CV-0109351; AMD-CV-0109381; AMD-CV-0109392; AMD-CV-0109404; AMD-CV-0109417; AMD-CV-0109837; AMD-CV-0109902; AMD-CV-0109906; AMD-CV-0110685; AMD-CV-0110688; AMD-CV-0110693; AMD-CV-0110761; AMD-CV-0110786; AMD-CV-0110802; AMD-CV-0110817; CV-AMD0000803; CV-AMD0000807; CV-AMD0000815; CV-AMD0000819; CV-AMD0000827; CV-AMD0000835; CV-AMD0000843; CV-AMD0000847; CV-AMD0000864; CV-AMD0000870; CV-AMD0000885; CV-AMD0000890; CV-AMD0001280; CV-AMD0001482.

Additional responsive agreements are being produced to Plaintiffs.

DATED: February 13, 2026

SHEPPARD, MULLIN, RICHTER & HAMPTON LLP

By /s/ *Harper S. Batts*

HARPER S. BATTS
CHRISTOPHER S. PONDER
JEFFREY LIANG
LAI YIP
Attorneys for Advanced Micro Devices, Inc. and Pensando Systems, Inc.

CERTIFICATE OF SERVICE

I HEREBY CERTIFY that on this day, February 13, 2026, a true and correct copy of the foregoing document, **DEFENDANTS ADVANCED MICRO DEVICES, INC. and PENSANDO SYSTEMS, INC.'s INVALIDITY CONTENTIONS AND ACCOMPANYING DOCUMENT PRODUCTION** was served by the indicated means to the persons at the addresses below:

Michael Matulewicz-Crowley Christine E. Lehman Adam Adler Courtland L. Reichman Jennifer Estremera Laura Carwile Scott Cole Ariane S. Mann **REICHMAN JORGENSEN LEHMAN & FELDBERG LLP**	***Counsel for Plaintiffs Concurrent Ventures, LLC and XtreamEdge, Inc.*** ***Via Email:*** mmatulewicz-crowley@reichmanjorgensen.com clehman@reichmanjorgensen.com aadler@reichmanjorgensen.com creichman@reichmanjorgensen.com jestremera@reichmanjorgensen.com lcarwile@reichmanjorgensen.com scole@reichmanjorgensen.com amann@reichmanjorgensen.com

Dated: February 13, 2026

By /s/ *Harper S. Batts*
HARPER S. BATTS

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