

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CAVIUM, INC.,

Petitioner,

v.

ALACRITECH INC.,

Patent Owner

Case IPR2017-01729
U.S. Patent 8,805,948

**PATENT OWNER'S PRELIMINARY RESPONSE
PURSUANT TO 35 U.S.C. § 313 AND 37 C.F.R. § 42.107**

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	1
II. OVERVIEW OF THE '948 PATENT	4
A. The '948 Patent Specification	4
B. The '948 Patent Claims	9
III. PROSECUTION HISTORY OF THE '948 PATENT	15
IV. OVERVIEW OF THE ASSERTED PRIOR ART	16
A. Thia, A Reduced Operation Protocol Engine (ROPE) for a Multiple-layer Bypass Architecture (1995) ("Thia")	17
B. Tanenbaum, Computer Networks, 3rd ed. (1996) ("Tanenbaum")	21
C. Stevens, TCP/IP Illustrated Volume 2: The Implementation ("Stevens") (Ex. 1063)	24
V. CLAIM CONSTRUCTION	25
VI. PETITIONER HAS NOT MADE A SUFFICIENT THRESHOLD SHOWING THAT STEVENS IS AVAILABLE AS PRIOR ART IN THIS PROCEEDING	26
VII. PETITIONER HAS NOT SUFFICIENTLY SHOWN A MOTIVATION TO COMBINE THIA WITH TANENBAUM OR STEVENS	33
VIII. PETITIONER HAS NOT SHOWN A DISCLOSURE OF PACKET PROCESSING FOR THE "CHECKING" LIMITATIONS OF THE CHALLENGED CLAIMS	35
IX. THE BOARD SHOULD DENY THE PETITION BECAUSE IT FAILS TO DISCLOSE ALL REAL PARTIES IN INTEREST	43
A. The Relationship Between Cavium and QLogic Is Sufficiently Close	44
B. Cavium Effectively Controls QLogic	45
C. Cavium Effectively Controls Dell	46
D. Dell Desires Review of the '948 Patent	48

E.	Intel Has Effective Choice as to the Legal Theories and Proofs of Dell and Cavium.....	49
F.	Finding QLogic, Dell, and Intel Are Real Parties in Interest Is Consistent with Legislative Intent.....	50
X.	ALACRITECH RESERVES ITS RIGHTS UNDER THE PENDING OIL STATES CASE AT THE UNITED STATES SUPREME COURT	51
XI.	THE BOARD SHOULD DECLINE INSTITUTION UNDER 35 U.S.C. § 325(D) BECAUSE THE PETITION RELIES ON PRIOR ART ALREADY CONSIDERED BY THE OFFICE	52
XII.	CONCLUSION	53

TABLE OF AUTHORITIES

Cases

Alacritech, Inc. v. CenturyLink, Inc.,
2:16-cv-00693-JRG-RSP (E.D. Tex.) 1, 43

Alacritech, Inc. v. Wistron Corp.,
2:16-cv-00692-JRG-RSP (E.D. Tex.) 1, 43

Alacritech, Inc. v. Dell Inc.,
2:16-cv-00695-RWS-RSP (E.D. Tex.)..... 1, 43

Beckman Instruments v. LKB Produkter AB,
892 F.2d 1547, 1551, 13 USPQ2d 1301, 1304 (Fed. Cir. 1989) 17

Benson & Ford, Inc. v. Wanda Petroleum Co.,
833 F.2d 1172 (5th Cir. 1987)..... 48

*Coalition for Affordable Drugs (ADROCA) LLC v. Acorda
Therapeutics, Inc.*, Case IPR2015-00817,
Paper, 12 (Aug. 24, 2015) 27

Cultec, Inc. v. Stormtech LLC,
IPR2017-00777, Paper 7 at 13 53

Dynamic Drinkware LLC v. Nat’l Graphics, Inc.,
800 F.3d 1375 (Fed. Cir. 2015)..... 26

Hewlett-Packard Co. v. U.S. Philips Corp.,
Case IPR2015-01505, Paper, 16 (Jan. 19, 2016) 26, 27

Microsoft Corp. v. Biscotti,
IPR2014-01457, Paper, No. 9 (Mar. 9, 2015)..... 27

Oil States Energy Servs. LLC v. Greene’s Energy Group, LLC,
Case No. 16-712, *certiorari granted* (U.S. Jun. 12, 2017) 51

Symbol Techs. Inc. v. Opticon Inc.,
935 F.2d 1569, 19 USPQ2d 1241 (Fed. Cir. 1991)..... 17

Teva Pharmaceuticals USA, Inc. v. Indivio UK Limited,
Case IPR2016-00280, Paper, 23 (Jun. 10, 2016) 27

Unified Patents Inc. v. John L. Berman,
IPR2016-01571, Paper 12 at 2-4 53

Statutory Authorities

35 U.S.C. § 103..... 17
35 U.S.C. § 103(a) 1
35 U.S.C. § 312(a)(2)..... 3, 43
35 U.S.C. § 313..... 1
35 U.S.C. § 314..... 3
35 U.S.C. § 314(a) 53
35 U.S.C. § 325(d)..... 52

Rules and Regulations

37 C.F.R. § 42.6(C)..... 21
37 C.F.R. § 42.6(e) 56
37 C.F.R. § 42.24..... 55
37 C.F.R. § 42.24(a)(i)..... 55
37 C.F.R. § 42.107(a) 1
37 C.F.R. § 42.108..... 3
37 CFR § 42.8(b)(1)..... 3, 43
Office Patent Trial Practice Guide,
77 Fed. Reg. 48756, 48759-60 (Aug. 14, 2012) 44, 51

Legislative Materials

157 Cong. Rec. S1034, S1041 (Mar. 1, 2011)..... 50
H.R. Rept. No. 112-98 (2011) (Judiciary Committee
Report on H.E. 1249, June 1, 2011) 50

Additional Authorities

Andrew S. Tanenbaum, *Computer Networks* (3rd ed. 1996) 2, 40
W. Richard Stevens and Gary R. Wright,
TCP/IP Illustrated Volume 2: The Implementation 2
Y.H. Thia and C.M. Woodside, i (1995) 2

PATENT OWNER'S LIST OF EXHIBITS

Exhibit #	Description
Ex. 2001	<i>Not used</i>
Ex. 2002	Intel Corporation's Motion to Intervene, Case No. 2:16-cv-00693-JRG-RSP, <i>Dkt.</i> 71 (E.D. Tex., Oct. 31, 2016)
Ex. 2003	Declaration of Karineh Khachatourian ISO Cavium's Motion to Intervene, Case No. 2:16-cv-00693-JRG-RSP, <i>Dkt.</i> 109-3 (E.D. Tex., Jan. 13, 2017).
Ex. 2004	<i>Not used</i>
Ex. 2005	<i>Not used</i>
Ex. 2006	<i>Not used</i>
Ex. 2007	<i>Not used</i>
Ex. 2008	<i>Not used</i>
Ex. 2009	Cavium's Motion to Intervene, Case No. 2:16-cv-00693-JRG-RSP, <i>Dkt.</i> 109 (E.D. Tex., Jan. 13, 2017)

I. INTRODUCTION

Pursuant to 35 U.S.C. § 313 and 37 C.F.R. § 42.107(a), Patent Owner Alacritech Inc. (“Alacritech”) submits this Preliminary Response to the Petition for *Inter Partes* Review (“the Petition”) filed in this matter. Petitioner Intel Corporation (“Intel”) seeks *Inter Partes* Review (“IPR”) of claims 1, 3, 6-9, 11, 14-17, 19, and 21-22 of U.S. Patent No. 8,805,948 (“the ’948 patent”), as allegedly being unpatentable under 35 U.S.C. § 103(a).¹

The ’948 patent is directed to accelerated network processing using an intelligent network interface card (INIC). Through a series of novel improvements over conventional network interfaces, the claimed invention is able to provide a fast-path that avoids protocol processing by the host for most large multipacket messages. As explained in more detail below, by relieving the host CPU of protocol processing for such messages, the claimed invention provides enhanced network and system performance, faster data throughput, increased system

¹ The ’948 patent is assigned to Alacritech and is the subject of co-pending litigation, *Alacritech, Inc. v. CenturyLink, Inc.*, 2:16-cv-00693-JRG-RSP (E.D. Tex.); *Alacritech, Inc. v. Wistron Corp.*, 2:16-cv-00692-JRG-RSP (E.D. Tex.); and *Alacritech, Inc. v. Dell Inc.*, 2:16-cv-00695-RWS-RSP (E.D. Tex.), which were all consolidated for pre-trial purposes (“the Litigations”).

stability, and an overall better user experience.

In its Petition, Cavium asserts that the '948 patent is invalid on a single ground—that claims 1, 3, 6-9, 11, 14-17, 19, and 21-22 of the '948 patent are obvious over Y.H. Thia and C.M. Woodside, i (1995) (“Thia”) (Ex. 1015), Andrew S. Tanenbaum, *Computer Networks* (3rd ed. 1996) (“Tanenbaum”) (Ex. 1006), and W. Richard Stevens and Gary R. Wright, *TCP/IP Illustrated Volume 2: The Implementation* (“Stevens”) (Ex. 1063). As set forth below, Cavium’s Petition does not establish a likelihood of success against any challenged claim.

First, the Board should deny institution because the Stevens reference—relied upon by Petitioner for every challenged claim—has not been shown to be available prior art in this proceeding. Specifically, Petitioner has not carried its institution-stage burden of producing sufficient admissible evidence that Stevens was publicly accessible prior to the October 14, 1997 priority date of the '948 patent and therefore qualifies as a “printed publication” here. In fact, Petitioner's own evidence states that the Stevens reference it submitted in this proceeding was publicly available no earlier than September 2010. This affects the entirety of this Petition, and the appropriate remedy is to deny institution.

Second, Petitioner has not sufficiently shown a motivation to combine Thia with Tanenbaum or Stevens. Although Petitioner refers to conditions in Tanenbaum’s discussion of header prediction and argues these conditions may be

implemented in a hardware bypass chip disclosed in Thia, Tanenbaum teaches against hardware implementations. This deficiency infects the entire Petition, and the appropriate remedy is to deny institution.

Third, the Petition identifies TPDU processing, not packet processing, in its obviousness analysis of the “check[ing]” limitations that appear in all challenged claims. The Petition offers no explanation as to why TPDUs are identified as meeting the “packet” limitations of the challenged claims, and according to Petitioner’s own references, TPDUs are not the same as packets with respect to the “check[ing]” limitations. As a result, the Petition does not adequately show that the claimed “check[ing] . . . the packets” limitations of the ‘948 patent appear in the relied-upon references. This deficiency affects all challenged claims, and should result in non-institution of the Petition in its entirety.

Fourth, the Board should deny institution because Petitioner has failed to identify all real parties in interest as required by 35 U.S.C. § 312(a)(2) and 37 CFR § 42.8(b)(1).

Accordingly, the references identified by Cavium do not give rise to a reasonable likelihood that it will prevail with respect to any challenged claim of the ‘948 patent, either alone or in combination. Therefore, the Board should not institute review on any claim of the ‘948 patent. *See* 35 U.S.C. § 314; 37 C.F.R. § 42.108.

II. OVERVIEW OF THE '948 PATENT

A. The '948 Patent Specification

The '948 patent describes a novel system for accelerating network processing. An intelligent network interface card (INIC) communicates with a host computer. The INIC provides “a fast-path that avoids host protocol processing for most large multipacket messages, greatly accelerating data communication.” Ex. 1001 at Abstract. In some embodiments, the INIC “contains hardware circuits configured for protocol processing that can perform that specific task faster than a host CPU.” *Id.*

As explained in the background of the '948 patent, when a conventional network interface card prepares to send data from a first host to a second host, “some control data is added at each layer of the first host regarding the protocol of that layer, the control data being indistinguishable from the original (payload) data for all lower layers of that host.” *Id.* at 2:39-42. For example,

an application layer attaches an application header to the payload data and sends the combined data to the presentation layer of the sending host, which receives the combined data, operates on it and adds a presentation header to the data, resulting in another combined data packet. The data resulting from combination of payload data, application header, and presentation header is then passed to the session layer, which performs required operations including attaching a session header to the data and presenting the resulting combination

of data to the transport layer. This process continues as the information moves to lower layers, with a transport header, network header, and data link header and trailer attached to the data at each of those layers, with each step typically including data moving and copying, before sending the data as bit packets over the network to the second host.

Id. at 2:42-57.

This process of adding a layer header to the data from the preceding layer is sometimes referred to as “encapsulation” because the data and layer header is treated as the data for the immediately following layer, which, in turn, adds its own layer header to the data from the preceding layer. Each layer is generally not aware of which portion of the data from the preceding layer is the preceding layer header of user data; as such, each layer treats the data it receives from the preceding layer as some generic payload. *Id.*

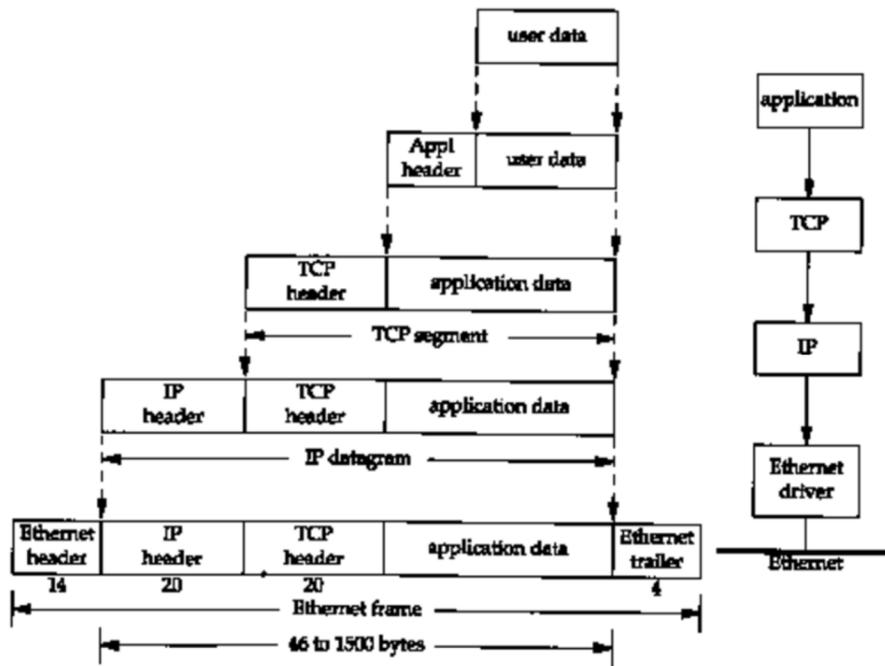


Figure 1.7 Encapsulation of data as it goes down the protocol stack.

Ex. 1008, Stevens at .034, Figure 1.7 (adapted from Petition at 19).

On the receiving side, the receiving host generally performs the reverse of the sending process, beginning with receiving the bit packets from the network. *Id.* at 2:58-60. Headers are removed, one at a time, and the received data is processed, in order, from the lowest (physical) layer to the highest (application) layer before transmission to a destination within the receiving host (*e.g.*, to the operating system space where the received data may be used by an application running on the receiving host). *Id.* at 2:60-63. Each layer of the receiving host recognizes and manipulates only the headers associated with that layer, since to that layer the higher layer header data is included with and indistinguishable from the payload data. *Id.* at 2:63-66. “Multiple interrupts, valuable central processing unit (CPU)

processing time and repeated data copies may also be necessary for the receiving host to place the data in an appropriate form at its intended destination.” *Id.* at 2:66-3:3.

The host CPU processes the data by constructing (transmit side) or destructing (receive side) the packet. The host CPU must be interrupted at least one time per layer and, in response, the host CPU processes each layer, which typically involves a copy and data manipulation operation (for example a checksum computation operation). An interrupt is a signal to the processor emitted by hardware or software indicating an event that needs immediate attention. An interrupt alerts the processor to a high-priority condition requiring the interruption of the current code the processor is executing. *Id.* This process involved with traditional network interface cards results in “repeated copying and interrupts to the CPU” of the host computer. *Id.* at 3:59-62. When the host CPU is interrupted, it generally must stop all other tasks it is currently working on, including tasks completely unrelated to the network processing. Frequent interrupts to the host CPU can be very disruptive to the host system generally and cause system instability and degraded system performance. *Id.*

The invention of the ’948 patent includes a “fast-path” where the host CPU is relieved of certain TCP/IP processing, which is instead performed by the INIC. In other words, the invention allows “data from the message to be processed via a

fast-path which accesses message data directly at its source or delivers it directly to its intended destination.” *Id.* at 3:53-57. The fast-path “bypasses conventional protocol processing of headers that accompany the data” and “employs a specialized microprocessor designed for processing network communication, avoiding the delays and pitfalls of conventional software layer processing, such as repeated copying and interrupts to the CPU.” *Id.* at 3:57-62.

The fast-path is shown in Figure 6 of the '948 patent, which is reproduced below. In this embodiment, the INIC performs at least the network and transport layer processing (e.g., IP and TCP layer processing in TCP/IP), freeing up the CPU on the host (“client”) computer to do other tasks. The fast-path also reduces or eliminates the number of interrupts sent to the CPU on the host/client. The more traditional “slow-path” is also shown, where the host/client is responsible for the IP and TCP layer processing. In the slow-path, the CPU on the host/client is interrupted at least one time for each layer for processing.

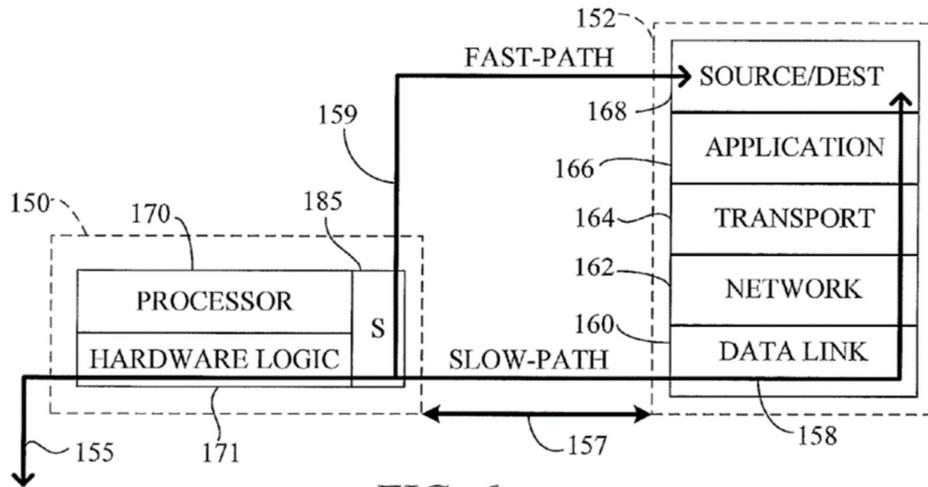


FIG. 6

Advantageously, the claimed invention allows for more efficient network processing by relieving the host CPU of per-frame processing and reducing the number of interrupts. *Id.* at 9:1-5. For fast-path communications, only one interrupt occurs at the beginning and end of an entire upper-layer message transaction, and “there are no interrupts for the sending or receiving of each lower layer portion or packet of that transaction.” *Id.* at 9:5-10. As stated above, the claimed arrangement allows for enhanced network and system performance, faster data throughput, increased system stability, and an overall better user experience.

B. The '948 Patent Claims

The '948 patent includes three independent claims. Independent claim 1 (reproduced below) recites a method for network communication. Notably, claim 1 recites, *inter alia*, “checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP

fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order.” This claimed feature is not found in any of the prior art cited by Petitioner.

Claim 1. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

running, on the host computer, a protocol processing stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, with an application layer running above the TCP layer;

initializing, by the host computer, a TCP connection that is defined by source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, first and second packets, wherein the first packet has a first TCP header and contains first payload data for the application, and the second packet has a second TCP header and contains second payload data for the application;

checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the first packet has any of the exception conditions, then protocol processing the first TCP header by the protocol processing stack;

if the second packet has any of the exception conditions, then protocol processing the second TCP header by the protocol processing stack;

if the packets do not have any of the exception conditions, then bypassing host protocol processing of the TCP headers and storing the first payload data and the second payload data together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the first payload data and the second payload data.

Id. at 19:43-20:7.

Claim 3 depends from claim 1 and additionally requires that a direct memory access (DMA) unit of the network interface performs the storing of the first payload data and the second payload data together in the buffer of the host computer. *Id.* at 20:8-10. Claim 6 also depends from claim 1 and adds the limitation that the network interface compares IP addresses and TCP ports of the packets with the source and destination IP addresses and source and destination TCP ports that define the TCP connection. *Id.* at 20:27-31. Claim 7 depends from claim 1 and recites that the checking of whether the packets have certain exception conditions includes checking whether the packets have a RST flag set. *Id.* at 20:32-34. Claim 8 depends from claim 1 and recites that the checking of whether the packets have certain exception conditions includes checking whether the packets have a SYN flag set. *Id.* at 20:35-37.

Independent claim 9 (reproduced below) recites a method for communicating information over a network. Notably, claim 9 recites, *inter alia*, “checking, by the network interface, whether the second and third packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order.” This claimed feature is not found in any of the prior art cited by Petitioner.

Claim 9. A method for network communication by a host computer having a network interface that is connected to the host by an input/output bus, the method comprising:

receiving, by the network interface, a first packet having a header including source and destination Internet Protocol (IP) addresses and source and destination Transmission Control Protocol (TCP) ports;

protocol processing, by the host computer, the first packet, thereby initializing a TCP connection that is defined by the source and destination IP addresses and source and destination TCP ports;

receiving, by the network interface, a second packet having a second header and payload data, wherein the second header has IP addresses and TCP ports that match the IP addresses and TCP ports of the TCP connection;

receiving, by the network interface, a third packet having a third header and additional payload data, wherein the third header has IP

addresses and TCP ports that match the IP addresses and TCP ports of the TCP connection;

checking, by the network interface, whether the second and third packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order;

if the second packet has any of the exception conditions, then protocol processing the second packet by the host computer;

if the third packet has any of the exception conditions, then protocol processing the third packet by the host computer;

if the second and third packets do not have any of the exception conditions, then storing the payload data of the second and third packets together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data of the second and third packets.

Id. at 20:38-21:7.

Claim 11 depends from claim 9 and additionally requires that a direct memory access (DMA) unit of the network interface performs the storing of the payload data of the second and third packets together in the buffer of the host computer. *Id.* at 21:12-15. Claim 14 also depends from claim 9 and adds the limitation that the network interface compares IP addresses and TCP ports of the second and third packets with the source and destination IP addresses and source

and destination TCP ports that define the TCP connection. *Id.* at 21:31-35. Claim 15 depends from claim 9 and recites that the checking of whether the second and third packets have certain exception conditions includes checking whether the packets have a RST flag set. *Id.* at 21:36-38. Claim 16 depends from claim 9 and recites that the checking of whether the second and third packets have certain exception conditions includes checking whether the packets have a SYN flag set. *Id.* at 21:39-41.

Independent claim 17 (reproduced below) recites an apparatus for network communication. Notably, claim 17 recites, *inter alia*, “check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order.” This claimed feature is not found in any of the prior art cited by Petitioner.

Claim 17. An apparatus for network communication, the apparatus comprising:

a host computer running a protocol stack including an Internet Protocol (IP) layer and a Transmission Control Protocol (TCP) layer, the protocol stack adapted to establish a TCP connection for an application layer running above the TCP layer, the TCP connection being defined by source and destination IP addresses and source and destination TCP ports;

a network interface that is connected to the host computer by an input/output bus, the network interface adapted to parse the headers

of received packets to determine whether the headers have the IP addresses and TCP ports that define the TCP connection and to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order, the network interface having logic that directs any of the received packets that have the exception conditions to the protocol stack for processing, and directs the received packets that do not have any of the exception conditions to have their headers removed and their payload data stored together in a buffer of the host computer, such that the payload data is stored in the buffer in order and without any TCP header stored between the payload data that came from different packets of the received packets.

Id. at 22:1-25. Claim 19 depends from claim 17 and recites that the network interface includes a direct memory access (DMA) unit adapted to store the payload data in the buffer of the host computer. *Id.* at 22:28-30. Claim 21 also depends from claim 17 and recites that the exception conditions include having a RST flag set. *Id.* at 22:36-37. Claim 22 depends from claim 17 and recites that the exception conditions include having a SYN flag set. *Id.* at 22:38-39.

III. PROSECUTION HISTORY OF THE '948 PATENT

The '948 patent issued on August 12, 2014. It was filed on September 26, 2013 as Application No. 14/038,297, which was a continuation of Application No. 09/692,561, filed October 18, 2000, which was a continuation of Application No.

09/067,544, filed April 27, 1998, which claims the benefit of Provisional Application No. 60/061,809, filed on October 14, 1997.

The '948 patent was subject to a thorough examination by Examiner Moustafa M. Meky, who allowed the application on June 20, 2014 after considering references (including Thia and Tanenbaum) disclosed in by the Applicant and conducting the Examiner's own prior art search on June 16, 2014. Ex. 1002 at .111-.149. In connection with the allowed claims, the Examiner stated:

None of the prior art of record taken singularly or in combination teaches or suggests a network interface of a host computer for *checking whether received packets have a certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or out of order*; processing any of the received packet that have the exception conditions, and *storing payload data of the received packets that do not have any of the exception conditions in a buffer of the host computer and without any TCP header stored between the payload data of the received packets.*

Ex. 1002 at .117 (emphasis added). An Issue Notification was mailed on July 23, 2014.

IV. OVERVIEW OF THE ASSERTED PRIOR ART

As described above, Cavium relies on Thia, Tanenbaum, and Stevens. These references, each alone or in combination, fail to teach or suggest all the limitations recited in the claims of the '948 patent. Moreover, at least the Stevens

reference has not been established as prior art. For example, the references are completely silent as to bypass exception conditions that involve checking whether the packets are IP fragmented. The following sections summarize these references and underscore their respective shortcomings.

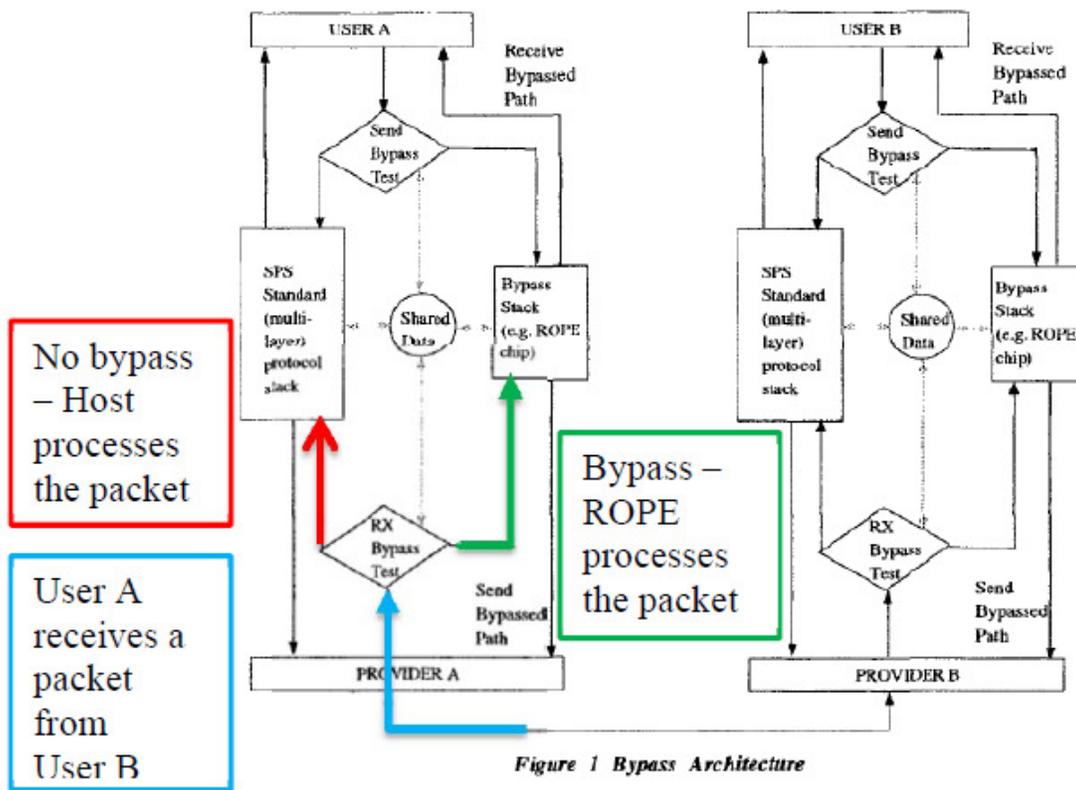
A. Thia, *A Reduced Operation Protocol Engine (ROPE) for a Multiple-layer Bypass Architecture* (1995) (“Thia”)

Thia appears on the face of the '948 patent under “References Cited” and was initialed by the Examiner in an Information Disclosure Statement (IDS) dated June 15, 2014. Ex. 1002 at .120-.144. Thia was therefore already considered by the Examiner during the prosecution of the '948 patent, which was found to be allowable over Thia.

As an initial matter, Thia presents a “feasibility study for a new approach to hardware assistance” and the “final step of generating a chip layout for fabrication and fault analysis was not performed.” Ex. 1015 at .002 and .008. Since Thia at best discloses an inoperative device, it is a non-enabling reference. A non-enabling reference is prior art under 35 U.S.C. § 103 “for all that it teaches.” *Beckman Instruments v. LKB Produkter AB*, 892 F.2d 1547, 1551, 13 USPQ2d 1301, 1304 (Fed. Cir. 1989); see *Symbol Techs. Inc. v. Opticon Inc.*, 935 F.2d 1569, 1578, 19 USPQ2d 1241, 1247 (Fed. Cir. 1991). The feasibility study discloses at a high level the idea of offloading session and transport layer

processing to a Reduced Operation Protocol Engine (“ROPE”) chip, but does not teach many of the implementation details.

This describes a bypass stack such as the ROPE chip that provides a hardware “fast path” for bulk data transfer. *Id.* at .002-.003.



Id. at .003, Fig. 1 (adapted from Petition at 35). Regarding the receive bypass test, This merely discloses a test that matches headers of incoming data packets with a template using header prediction:

2.1 Bypass Architecture

Figure 1 illustrates the architecture of a bypass implementation for any standard protocol. The standard protocol stack (SPS) is the processing path taken by all PDUs during a connection without the bypass. The SPS may refer to a single layer or to multiple adjoining layers of a layered protocol stack. The bypass has 4 key components:

- Send Bypass Test*;
- Receive Bypass Test*;
- Bypass Stack*;
- Shared data* for access by the two tests, the Bypass stack and the SPS.

The send bypass test identifies outgoing packets that are data packets in the data transfer phase. The receive bypass test matches the incoming PDU headers with a template that identifies the predicted bypassable headers. The bypass stack performs all the relevant protocol processing in the data transfer phase. The shared data are used to maintain state consistency between the SPS and the bypass stack, including window flow control parameters and connection identifiers. Whenever there is a change in the processing path between the SPS and the bypass stack, checks are performed to ensure that there are no outstanding packets in the current path, i.e. "no in-transit PDUs", before the change is made. A more detailed discussion on this is presented in another paper [33].

Id. at .003 (adapted from Petition at 34). Thia does not address exception conditions relating to the bypass test.

Table 1 of Thia “identifies procedures which are strong candidates for implementation in the bypass chip, and those which are better handled by the host, during the data transfer phase.” *Id.* at .006.

Layer	Procedure	Bypass Chip	Host	Per-Octet (A)	Per-Packet (B)	Per-Group-Of-Packets Aggregated to Per-Packet for bulk data transfer (B)	Remarks
Presentation	Encoding	X		X			
	Encryption	X		X			
	Compression	X		X			
	Context Alteration		X			X	
Session	Synchronization Management		X			X	
	Token management		X			X	
Transport (Class 4)	Checksum (Optional)	X		X			
	Timer Management	X			X	X	Depends on Implementation
	Generation of ACK packets (Flow Control)	X				X	
	Resequencing	X			X		
All 3 layers	Header Construction	X			X		
	Header Decode	X			X		
	Buffer Management	X			X		Minimized (Simple scheme)
	Context Switching	X			X	X	Moved away from host OS
	Data Copying	With multiple-layer bypass, data Copying within layers is eliminated.				X	Use of dual-ported memory and DMA..

Table 1 Bypassable versus Non-bypassable functions

Id. As shown in the table, This identifies bypassable functions in the presentation layer, the session layer, the transport layer, and all three of those layers. This does not address the bypass test or bypassable functions in relation to the network layer, which is where fragmentation/segmentation occurs. Indeed, This assumes that any packets that have been fragmented have been reassembled at the receiver before being passed to the Transport layer. *Id.* at .013-.014 (stating: “In an ATM system we assume that the segmentation and reassembly or SAR operation would also be in hardware, since it is done frequently”; “The Segmentation and Reassembly

sublayer of the ATM adaption layer is a good place for [segmentation/reassembly] functions”).

B. Tanenbaum, Computer Networks, 3rd ed. (1996) (“Tanenbaum”)²

Tanenbaum, “Computer Networks,” is the third edition of a textbook relating to computer networks. As acknowledged by Petitioner, it too was already considered by the Examiner during the prosecution of the ’948 patent, which was found to be allowable over Tanenbaum. Petition at n.6. On pages 583-586, Tanenbaum describes “fast” transport protocol data unit (“TPDU”) processing. Ex. 1006 at .583-86. On the sending side, Tanenbaum notes that “[i]n the normal case, the headers of consecutive data TPDU’s are almost the same.” *Id.* at .583. In view of this observation, a “prototype header” is defined. *Id.* In order to construct a packet for transmission, at the TCP layer, the TCP prototype header is copied into the output buffer, the sequence number is filled in, the TCP checksum is computed, and the sequence number is incremented in memory. *Id.* at .584. Then, the TCP header and data is handed “to a special IP procedure” at the IP layer,

² Petitioner has failed to file the portions of Tanenbaum cited in the Petition as an exhibit in this case in accordance with 37 C.F.R. § 42.6(C). Petitioner’s Exhibit 1006 only includes up to .496 of Tanenbaum, whereas Petitioner cites to pages of Tanenbaum after .496.

where the IP prototype header is copied into the output buffer, the “Identification” field is inserted, and the IP checksum is computed. *Id.* The packet is then ready for transmission.

On the receive side, a “connection record” is stored in a hash table for lookup (or the last record is tried first). *Id.* at .585. Then conditions for a special “fast path” TCP procedure are checked:

The TPDU is then checked to see if it is a normal one: the state is *ESTABLISHED*, neither side is trying to close the connection, the TPDU is a full one, no special flags are set, and the sequence number is the one expected. These tests take just a handful of instructions. If all conditions are met, a special fast path TCP procedure is called.

Id. at 585. The connection record is updated, and data is copied to the user. *Id.* During the copy process, the fast path computes the checksum. *Id.* Tanenbaum explains that this general scheme of first making a quick check to see if the header is what is expected, and having a special procedure to handle that case, is called “header prediction.” *Id.* However, Tanenbaum discloses conditions that relate to checking TPDU, *i.e.* transport protocol data units, and not packets. The test is performed at the transport layer and not the network layer where packets are processed to generate TPDU to be passed on to the transport layer.

Further, Tanenbaum teaches away from performing any TCP/IP protocol processing on anything other than the host CPU. Tanenbaum is aware of the

possibility of a transport entity being on a network interface card:

The hardware and/or software within the transport layer that does the work is called the **transport entity**. The transport entity can be in the operating system kernel, in a separate user process, in a library package bound into network applications, or on the network interface card.

Id. at .498 (bold in original). Petitioner points to this to assert that Tanenbaum “teaches that the transport entity may be built into the network interface card.”

Petition at 45. However, Tanenbaum clearly teaches away from doing so:

A tempting way to go fast is to build fast network interfaces in hardware. The difficulty with this strategy is that unless the protocol is exceedingly simple, hardware just means a plug-in board with a second CPU and its own program. To avoid having the network coprocessor be as expensive as the main CPU, it is often a slower chip. The consequence of this design is that much of the time the main (fast) CPU is idle waiting for the second (slow) CPU to do the critical work. It is a myth to think that the main CPU has other work to do while waiting. Furthermore, when two general-purpose CPUs communicate, race conditions can occur, so elaborate protocols are needed between the two processors to synchronize them correctly. Usually, the best approach is to make the protocols simple and have the main CPU do the work.

Id. at .588-89.

C. Stevens, *TCP/IP Illustrated Volume 2: The Implementation* (“Stevens”) (Ex. 1063)

Stevens is a reference book on the implementation of TCP/IP.³ Ex. 1013 at .023. Stevens discloses header prediction in which “[i]f TCP is receiving data, the next expected segment for this connection is the next in-sequence data segment. . . . [A] small set of tests determines if the next expected segment has been received, and if so, it is handled in-line, faster than the general processing.” *Id.* at .961. Steven discusses six conditions for a segment to be the next in-sequence data segment:

Check if segment is the next expected

347-366 The following six conditions must all be true for the segment to be the next expected data segment or the next expected ACK:

1. The connection state must be ESTABLISHED.
2. The following four control flags must not be on: SYN, FIN, RST, or URG. The ACK flag must be on. In other words, of the six TCP control flags, the ACK flag must be set, the four just listed must be cleared, and it doesn't matter whether

³ Stevens is not available as prior art in this proceeding. Although Petitioner alleges “Stevens[] was published no later than April 7, 1995 and is prior art under 102(b),” Petitioner has not provided sufficient evidence that Stevens qualifies as a “printed publication” against the challenged ’948 patent claims. *See infra* § VI.

PSH is set or cleared. (Normally in the ESTABLISHED state the ACK flag is always on unless the RST flag is on.)

3. If the segment contains a timestamp option, the timestamp value from the other end (`ts_val`) must be greater than or equal to the previous timestamp received for this connection (`ts_recent`). This is basically the PAWS test, which we describe in detail in Section 28.7. If `ts_val` is less than `ts_recent`, this segment is out of order because it was sent before the most previous segment received on this connection. Since the other end always sends its timestamp clock (the global variable `tcp_now` in Net/3) as its timestamp value, the received timestamps of in-order segments always form a monotonic increasing sequence.

The timestamp need not increase with every in-order segment. Indeed, on a Net/3 system that increments the timestamp clock (`tcp_now`) every 500 ms, multiple segments are often sent on a connection before that clock is incremented. Think of the timestamp and sequence number as forming a 64-bit value, with the sequence number in the low-order 32 bits and the timestamp in the high-order 32 bits. This 64-bit value always increases by at least 1 for every in-order segment (taking into account the modulo arithmetic).

4. The starting sequence number of the segment (`ti_seq`) must equal the next expected receive sequence number (`rcv_nxt`). If this test is false, then the received segment is either a retransmission or a segment beyond the one expected.
5. The window advertised by the segment (`tiwin`) must be nonzero, and must equal the current send window (`snd_wnd`). This means the window has not changed.
6. The next sequence number to send (`snd_nxt`) must equal the highest sequence number sent (`snd_max`). This means the last segment sent by TCP was not a retransmission.

Id. at .962-63 (adopted from Petition at 46-47). However, as acknowledged by the Petition, “[t]he code above is implemented at the TCP layer, and thus, does not check for IP fragmentation because the IP layer reassembles fragmented IP packets.” Petition at 47. Further, Stevens does not address offloading protocol processing to a network interface chip.

V. CLAIM CONSTRUCTION

The Petition does not assert that the challenged claims require express construction by the Board, and Patent Owner agrees.

VI. PETITIONER HAS NOT MADE A SUFFICIENT THRESHOLD SHOWING THAT STEVENS IS AVAILABLE AS PRIOR ART IN THIS PROCEEDING

Petitioner asserts only a single ground of patentability—obviousness over Thia, Tanenbaum, and Stevens—and relies on Stevens for disclosure in all of the challenged claims. (*See, e.g.*, Pet. at 66-69; 84; 95-96.)

Petitioner, however, fails has failed to make a sufficient threshold showing that Stevens is available as prior art in this proceeding, as required under *Dynamic Drinkware LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1379 (Fed. Cir. 2015) (“*Dynamic Drinkware*”) and its progeny. In fact, what evidence Petitioner has submitted tends to show that Stevens was publicly available no earlier than September 2010, more than a decade after the priority date of the challenged '948 patent claims. (*See* Ex. 1013.010 (stating “24th Printing September 2010”).) As a result, the Board should deny institution.

Under *Dynamic Drinkware*, “Petitioner has the initial burden of production of evidence establishing that asserted references are applicable as prior art.” *Hewlett-Packard Co. v. U.S. Philips Corp.*, Case IPR2015-01505, Paper 16 (Jan. 19, 2016), at 6 (citing *Dynamic Drinkware*, 800 F.3d at 1379). The Board “looks to the underlying facts to make a legal determination as to whether a document is a printed publication.” *Id.* (citation omitted). “The determination of whether a document is a ‘printed publication’ . . . involves a case-by-case inquiry into the

facts and circumstances surrounding its disclosure to members of the public.” *Id.*
(citations omitted).

More specific to Petitioner’s burden:

To qualify as a printed publication, a document must have been sufficiently accessible to the public interested in the art. . . . *The party seeking to introduce the reference should produce sufficient proof of its dissemination or that it has otherwise been available and accessible to persons concerned with the art to which the document relates and thus most likely to avail themselves of its contents.*

Id. at 7 (quotation marks and citations omitted; emphasis added). Without such a showing, institution should be denied. *See id.* at 6-11; *Coalition for Affordable Drugs (ADROCA) LLC v. Acorda Therapeutics, Inc.*, Case IPR2015-00817, Paper 12 (Aug. 24, 2015), at 5-6; *Teva Pharmaceuticals USA, Inc. v. Indivio UK Limited*, Case IPR2016-00280, Paper 23 (Jun. 10, 2016), at 8-11; *Microsoft Corp. v. Biscotti*, IPR2014-01457, Paper No. 9, at 25-28 (Mar. 9, 2015).

Such is the case here with respect to the Stevens. Stevens is potentially available as prior art in this proceeding only as a § 102(b) “printed publication” reference. In order to assuage its institution-stage “threshold burden” with respect to Stevens, Petitioner needed to put forth sufficient evidence that Stevens—and

more specifically, the edition of Stevens submitted as evidence here (Ex. 1013)—qualifies as printed publication prior art to the challenged claims. *See, e.g.*, IPR2015-01505, Paper 16, at 6-11; IPR2015-00817, Paper 12, at 5-6; IPR2016-00280, Paper 23, at 8-11. Petitioner has put forth no persuasive evidence on this front, and in fact has submitted evidence directly contradictory to its position on Stevens's prior art status.

As putative support for Stevens's prior art status, the Petition cites Ex. 1063, the Affidavit of Pamela Stansbury.⁴ The Stansbury Affidavit is of negligible evidentiary value here, because it is uncertain and conclusory—and devoid of details as to the basis for (and meaning of) Ms. Stansbury's uncertain conclusion. The entire substance of the Stansbury Affidavit is shown below:

⁴ Specifically, the Petition states, without more: “Stevens2 was published no later than April 7, 1995 and is prior art under 102(b). *See also* Ex. 1063 (declaration of librarian).” Pet. at 45 n9.

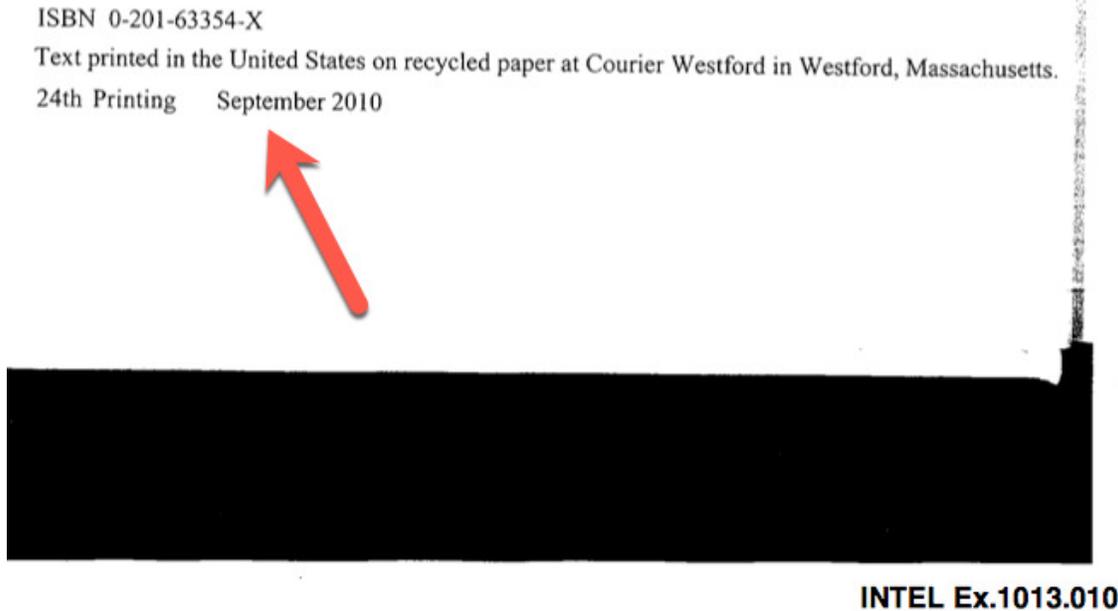
Pamela Stansbury being of full age and duly sworn, deposes and says as follows:

1. I am an employee of the Cornell University Library, and specifically the Original Cataloging Unit, located at Cornell University, Ithaca, New York 14853. I am familiar with the policies and procedures of the Library as they relate to the receipt, cataloging, and tracking of books and I have personal knowledge of the facts set forth below.
2. I am the Administrative Supervisor in the Original Cataloging Unit, which maintains bibliographical and processing information for many historical documents. I have held such position since 1996.
3. Included in the Library's historical collection are various publications. As part of that collection, the Library maintains custody of an original copy of **TCP/IP illustrated, volume 2 / W. Richard Stevens (Addison-Wesley Publishing, 1995)**
4. Weil, Gotshal & Manges LLP requested information about **TCP/IP illustrated, volume 2 / W. Richard Stevens** - specifically when this item was first made publicly available by the Library. As best I can determine, the publication was publicly available at the Cornell University Library as of April 7, 1995.

This is the entirety of Ms. Stansbury's opinion, and is wholly insufficient, as it provides no meaningful evidence as to Steven's public availability. First, the Affidavit has absolutely no explanation or details as to what Ms. Stansbury means when she says a "publicly available." Moreover, there is no evidence or explanation as to how Ms. Stansbury attempted to determine Stevens's putative public availability. It is clear from the Affidavit that Ms. Stansbury has no personal knowledge of Stevens's public availability, yet there's a complete absence of any explanation of the methodology Ms. Stansbury used in lieu of personal knowledge. Moreover, there's no explanation of Ms. Stansbury's hedging language, and why she could only state "as best I can determine . . ." in her Affidavit.

In short, the Stansbury Affidavit offers no details or background from which the Board can reliably determine the meaning, *vis-à-vis* the "printed publication" standard, of Ms. Stansbury's conclusory statement that Stevens was "publicly available." And Ms. Stansbury's conclusory statement is itself uncertain: we don't know from the Affidavit what "as best I can determine" means, and this is a serious problem from an evidentiary perspective.

In any event, other evidence in the record as to the public availability of Ex. 1013—the putative Stevens reference cited and relied upon in the Petition—affirmatively indicates that the Stevens reference submitted in this proceeding is not prior art. The Stevens reference that Petitioner relies on (Ex. 1013) is, on its face, a “24th printing September 2010,” indicating that the actual scanned document submitted as putative prior art in this proceeding was not publicly available prior to at least 2010. (Ex. 1013.010)



(*Id.* (arrow added).)

Without some kind of authenticating or explanatory evidence in the record as to Steven's distribution, a bare copyright notice does not sufficiently assuage Petitioner's burden of production as to Stevens's "printed publication" prior art status—and the "September 2010" legend strongly argues against such availability for the reference actually submitted as evidence here. As the Board explained in Case IPR2015-01505:

On its face Chambers includes a "December 1985" date, as well as a "BBC RD 1985/15" indication. . . . The fact that a date is printed on the face of a reference, without more, is not enough to establish that the reference was publicly accessible on that date. *See LG Elecs., Inc. v. Advanced Micro Devices, Inc.*, Case IPR2015-00329, slip

op. at 13 (PTAB July 10, 2015) (Paper 13); *see also* *Google Inc. v. ART+COM Innovationpool GmbH*, Case IPR2015-00788, slip op. at 8 (PTAB Sept. 2, 2015) (Paper 7) (“[T]his bare date, without more, does not provide any information about the date [the reference] was publicly accessible.”) Further the unsupported testimony of Dr. Jayan that “[Chambers] was published in December 1985” also is insufficient to demonstrate that Chambers was published in December 1985.

IPR2015-01505, Paper 16, at 8 (footnotes and exhibit citations omitted; brackets and alterations in original).

Indeed, even if some version of Stevens was printed in 1995, that would still not satisfy Petitioner’s burden, as the version on which it relies is the 24th printing. There is no indication—not even the barest of allegations—that the 24th printing is the same as the one that was allegedly available in 1995. Without such information, it is impossible to rely on the 1995 copyright date or Ms. Stansbury’s conclusory declaration. Indeed, the remainder of the Petition does not provide any further explanation (not even attorney argument) as to the Stevens reference's prior art status.

In sum, the Petition fails to sufficiently establish the Stevens reference's availability as prior art to the challenged claims. Since Stevens is integral to all challenged claims in Ground 1, the Petition should be denied in its entirety. *See*

IPR2015-01505, Paper 16, at 6-11; IPR2015-00817, Paper 12, at 5-6; IPR2016-00280, Paper 23, at 8-11.

VII. PETITIONER HAS NOT SUFFICIENTLY SHOWN A MOTIVATION TO COMBINE THIA WITH TANENBAUM OR STEVENS

The Petition fails to establish that a person skilled in the art would have had the requisite motivation to combine Thia with either Tanenbaum or Stevens at the time of the challenged '948 patent claims. This deficiency affects all challenged claims, and the Board should deny institution for this additional reason.

Petitioner asserts that one skilled in the art reading Thia in combination with Tanenbaum would have been motivated to combine because “both are based on Jacobson’s TCP/IP Header Prediction,” and one skilled in the art would have been motivated “to reduce the burden of protocol processing on the host processor such that it may perform other necessary functions.” (Pet. at 50.) However, as discussed above in Section IV, Tanenbaum clearly teaches away from implementing Tanenbaum’s conditions in hardware such as Thia’s ROPE chip:

A tempting way to go fast is to build fast network interfaces in hardware. The difficulty with this strategy is that unless the protocol is exceedingly simple, hardware just means a plug-in board with a second CPU and its own program. To avoid having the network coprocessor be as expensive as the main CPU, it is often a slower chip. The consequence of this design is that much of the time the main (fast) CPU is idle waiting for the second (slow) CPU to do the

critical work. It is a myth to think that the main CPU has other work to do while waiting. Furthermore, when two general-purpose CPUs communicate, race conditions can occur, so elaborate protocols are needed between the two processors to synchronize them correctly. Usually, the best approach is to make the protocols simple and have the main CPU do the work.

(*Id.* at .588-89.) Moreover, Thia was merely a “*feasibility study* for a new approach to hardware assistance” and “the final step of generating a chip layout for fabrication and fault analysis was not performed.” (Ex. 1015 at .002 and .008.) Contrary to Petitioner’s assertions, one skilled in the art reading Thia in combination with Tanenbaum would not be motivated to ignore or work against the teachings of Tanenbaum to attempt Tanenbaum’s conditions for header predictions in combination with the theoretical chip discussed in Thia.

Regarding the motivation to combine Stevens with Thia, Petitioner conclusorily asserts that one skilled in the art would have been motivated to combine the receive bypass test in Thia with the conditions of Jacobson’s Header Prediction explained in Stevens because one skilled in the art “would want to reduce the burden of protocol processing on the host processor to reduce the bottleneck recognized by Thia and Tanenbaum.” (Pet. at 51.) However, as the Petition itself acknowledges, Stevens’s discussion of header prediction is directed to host processes, and does not address offloading protocol processing to a network

interface chip. (*See* Pet. at 47.) Moreover, as illustrated by Tanenbaum’s teaching away from implementing protocol processing in hardware, it would not have been obvious to one skilled in the art to offload the processes disclosed in Stevens onto a hardware chip. Accordingly, one skilled in the art would not have been motivated to combine Thia with Stevens.

Petitioner’s assertions that one skilled in the art would be motivated to combine Thia with Tanenbaum or Stevens are based on improper hindsight reasoning. Indeed, Thia’s “feasibility study for a new approach to hardware assistance,” which was theoretical in nature, illustrates that even professors at research universities were uncertain of the feasibility of offloading parts of multi-layer protocol stack processing to hardware. (Ex. 1015 at .002.)

In view of the above, the Petition does not establish that it would have been obvious to one skilled in the art to combine Thia’s theoretical research regarding a ROPE chip with the header prediction disclosed by Tanenbaum and Stevens. This deficiency affects all challenged claims, and the Board should decline institution for this reason alone.

VIII. PETITIONER HAS NOT SHOWN A DISCLOSURE OF *PACKET* PROCESSING FOR THE "CHECKING" LIMITATIONS OF THE CHALLENGED CLAIMS

The Petition should be denied in its entirety for an additional reason: the challenged claims require that “packets” undergo certain types of processing, but

the Petition only identifies “TPDU” or “segment” processing in the alleged prior art. The Petition doesn’t explain the difference (if any) between the claimed “packets” and the disclosed “TPDUs”/“segments,” and Petitioner’s own references state that the two are not the same. In short, Petitioner has failed to meet its obviousness burden of showing that all limitations of the challenged claims are disclosed in the relied upon references. Institution should be denied on this basis.

The challenged claims include (among others), the following limitations that require specified processing of “packets”:

- “checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order” (claim 1)
- “checking, by the network interface, whether the second and third packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order” (claim 9)
- "to check whether the packets have certain exception conditions, including whether the packets are IP fragmented, have a FIN flag set, or are out of order" (claim 17)

The above “check[ing]” limitations—which reside in all challenged claims—specifically recite “packets,” and Petitioner does not argue that “packets” should be given any special construction. (*See* Pet. at 27.) Yet the Petition itself does not identify packets being processed in the required manner. (*See* Pet. at 54-55.) Instead, the Petition identifies the putative exception conditions being checked for “TPDUs” (Tanenbaum) and “segments” (Stevens). (*Id.*) Specifically, the Petition identifies only the following disclosures in its entire analysis of the “check[ing]” limitations:

- Thia (Ex. 1015) at 002, 014
- Tanenbaum (Ex. 1006) at 545, 583-585
- Stevens (Ex. 1013) at 962-63

(*See* Pet. at 54-55, 69, 80.)⁵

⁵ Section 10.11.3.4 of the Petition (Pet. at 80) refers back to pages 66-67 for analysis of the “to check” limitation of claim 17, but cites page 567 of Tanenbaum and pages 936-37 of Stevens as putative support. (*See* Pet. at 95-96.) These (different) citations appear to be typographical errors, since they refer, respectively, to (i) a page in the middle of a lengthy discussion of ATM Adaptation Layer protocols in Tanenbaum (Ex. 1006.567, *cited in* Pet. at 80), and (ii) two

As explained below, these citations do not collectively disclose “checking, by the network interface, whether the packets have certain exception conditions, including checking whether the packets are IP fragmented, checking whether the packets have a FIN flag set, and checking whether the packets are out of order,” and related limitations in claims 9 and 17. As a result, the Petition fails on its face to show disclosure of all challenged limitations in the alleged prior art.

First, pages 2 and 14 of Thia, cited on pp. 54-55 of the Petition, do not disclose any “check[ing] . . . whether the packets have any exception conditions” of any kind. Indeed, neither portion of Thia so much as mentions exception conditions of any sort. (*See* Ex. 1015.002, 014.)

Next, pages 545 and 583-85 of Tanenbaum, cited on p. 55 of the Petition, do not disclose the claimed “check[ing] . . . whether the packets have any exception conditions.” As the Petition concedes, page 545 of Tanenbaum does not disclose “checking” of any sort. (*See* Pet. 55; Ex. 1006.545.) Moreover, pages 583-85 of Tanenbaum—the primary alleged disclosure of the claimed “check[ing]” limitations describes, on its face, only “Fast TPDU Processing”:

pages in the middle of a multi-page discussion of the tcp_reass Function in Stevens (Ex. 1013.936-37, *cited in* Pet. at 80.).

6.6.4. Fast TPDU Processing

The moral of the story above is that the main obstacle to fast networking is protocol software. In this section we will look at some ways to speed up this software. For more information, see (Clark et al., 1989; Edwards and Muir, 1995; and Chandranmenon and Varghese, 1995).

TPDU processing overhead has two components: overhead per TPDU and overhead per byte. Both must be attacked. The key to fast TPDU processing is to separate out the normal case (one-way data transfer) and handle it specially. Although a sequence of special TPDUs are needed to get into the *ESTABLISHED* state, once there, TPDU processing is straightforward until one side starts to close the connection.

Let us begin by examining the sending side in the *ESTABLISHED* state when there are data to be transmitted. For the sake of clarity, we assume here that the transport entity is in the kernel, although the same ideas apply if it is a user-space process or a library inside the sending process. In Fig. 6-49, the sending process traps into the kernel to do the SEND. The first thing the transport entity does is make a test to see if this is the normal case: the state is *ESTABLISHED*, neither side is trying to close the connection, a regular (i.e., not an out-of-band) full TPDU is being sent, and there is enough window space available at the receiver. If all conditions are met, no further tests are needed and the fast path through the sending transport entity can be taken.

In the normal case, the headers of consecutive data TPDUs are almost the same. To take advantage of this fact, a prototype header is stored within the transport entity. At the start of the fast path, it is copied as fast as possible to a scratch buffer, word by word. Those fields that change from TPDU to TPDU are then overwritten in the buffer. Frequently, these fields are easily derived from state variables, such as the next sequence number. A pointer to the full TPDU header plus a pointer to the user data are then passed to the network layer. Here the same strategy can be followed (not shown in Fig. 6-49). Finally, the network layer gives the resulting packet to the data link layer for transmission.

(Ex. 1006.583; *see also id.* at 584-86 (reciting only TPDU processing).)

Thus, the claims require “checking . . . whether the packets have any exception conditions,” but the identified portions of Tanenbaum disclose (if anything) “checking . . . whether TPDUs have exception conditions.” The Petition provides no explanation of why TPDUs meet the claimed “packet” requirement—and indeed tries to hide the distinction by inserting, without explanation or notice, the following bracketed material into a block quote from Tanenbaum 584-85:

Now let us look at fast path processing on the receiving side.... For TCP, the connection record can be stored in a hash table for which some simple function of the two IP addresses and two ports is the key. Once the connection record has been located, both addresses and both ports must be compared to verify that the correct record has been found.... the TPDU [Transport Protocol Data Unit, i.e. packet] is then checked

to see if it is a normal one: the state is *ESTABLISHED*, neither side is trying to close the connection, the TPDU is a full one, no special flags are set,⁴ and the sequence number is the one expected. These tests take just a handful of instructions. If all conditions are met, a special fast path TCP procedure is called.

(Pet. at 33-34 (quoting Ex. 1006 at 584-85).)⁶

Critically, not only does the Petition fail to explain why the relied-upon “TPDUs” in Tanenbaum meet the required “packets” limitations in the challenged claims, Tanenbaum itself states that “TPDUs” are not the same thing as “packets.” Tanenbaum explains that protocol data units at the network layer (e.g., IP layer in TCP/IP) are called “packets.” (*See, e.g.,* Ex. 1006.047, Fig. 1-16.) Protocol data units at the transport layer (e.g., TCP layer in TCP/IP) are called TPDUs (*i.e.* transport protocol data units). (*Id.*) As described by Tanenbaum, TPDUs refer to

⁶ The Petition’s citation for the above block quote does not signal or otherwise warn the reader that the bracketed material equating “TPDU” with “packet” was added by Intel’s lawyers and does not appear in Tanenbaum.

“messages sent from transport entity to transport entity” and “[t]hus TPDUs (exchanged by the transport layer) are contained in packets (exchanged by the network layer).” (Ex. 1006.543 (emphasis added).) This is illustrated in the Figure below from Tanenbaum.

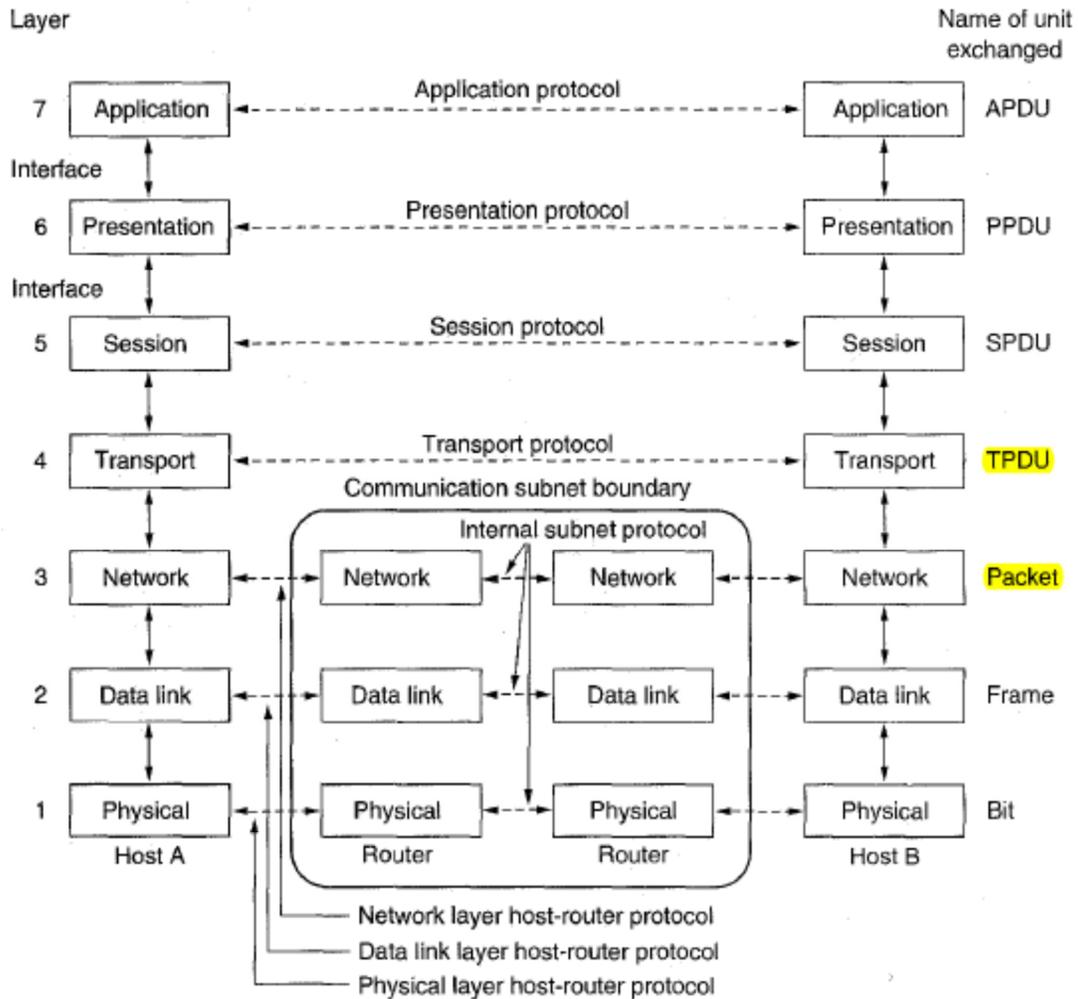


Fig. 1-16. The OSI reference model.

(Ex. 1006.047, Fig. 1-16 (emphasis added).)

In short, the relied-upon portions of Tanenbaum—which relate to the

processing of TPDU—cannot meet Petitioner’s burden of showing that “check[ing] . . . whether the packets have any exception conditions” is found in the prior art.

Finally, Petitioner cites pages 962-63 of Stevens. (Pet. at 55.) But this portion of Stevens (which, as noted earlier, has not been shown to be prior art in any event) does not disclose any type of “check[ing] . . . whether the packets have any exception conditions” either. The relied-upon portions of Stevens expressly and exclusively discuss “segment”—not “packet”—processing. (*See* Ex. 1013.964-65.) The Petition doesn’t explain why Stevens’s “segment” can meet the “packet” limitations of the challenged claims. (*See, e.g.*, Pet. at 54-55.) And more to the point, the Petition actually explains why Stevens’s “segment” can’t meet the “check[ing] . . . whether the packets have any exception conditions” limitations: as the Petition explains the cited code from Stevens “is implemented at the TCP layer, and thus, does not check for IP fragmentation because the IP layer reassembles fragmented IP packets.” (Pet. at 40.) In short, the Petition itself states that Stevens doesn’t disclose the “check[ing]” limitations.

In view of the above, the Petition does not adequately identify any disclosure of the “check[ing] . . . whether the packets have any exception conditions” limitations—which appear in all challenged claims—in any of the relied upon references. For at least this reason, the Petition should be denied in its entirety.

IX. THE BOARD SHOULD DENY THE PETITION BECAUSE IT FAILS TO DISCLOSE ALL REAL PARTIES IN INTEREST

A requirement of a petition includes identifying “all real parties in interest.” 35 U.S.C. § 312(a)(2).⁷ The Board should deny institution of this IPR because the Petition fails to identify all real parties in interest as required by 35 U.S.C. § 312(a)(2) and 37 CFR § 42.8(b)(1).

The Petition identifies only one real party in interest: Cavium.⁸ In doing so, the Petitioner at least fails to identify at least QLogic Corporation (“QLogic”)—a subsidiary of Cavium being a supplier of defendant Dell Inc. (“Dell”) in Alacritech’s patent infringement lawsuit over the ’948 Patent.⁹ Cavium also failed

⁷ See also IPR2014-00689, Paper No. 22 (Aug. 12, 2014) (holding that failing to list a real party in interest can result in the petition to be considered “an incomplete petition” under Rule 42.106(b), not entitled to receive a filing date).

⁸ See Petition at 2 (“Petitioner, Cavium, Inc., is the real-party-in-interest. No other parties exercised or could have exercised control over this petition; no other parties funded or directed this Petition.”).

⁹ The co-pending district court cases, *Alacritech v. CenturyLink*, *Alacritech v. Wistron*, and *Alacritech v. Dell*, were filed on June 30, 2016, and consolidated for pre-trial purposes. In January 2017, Cavium filed a motion to intervene in the district court case because Dell’s accused products used components supplied by

to disclose the district court defendant Dell, and another indemnifier Intel, the *de facto* driver of this case.

A. The Relationship Between Cavium and QLogic Is Sufficiently Close

“Privity is essentially a shorthand statement that collateral estoppel is to be applied in a given case. . . . The concept refers to a relationship between the party to be estopped and the unsuccessful party in the prior litigation which is sufficiently close so as to justify application of the doctrine of collateral estoppel.” Office Patent Trial Practice Guide, 77 Fed. Reg. at 48759.

Cavium is QLogic’s parent company and QLogic is Cavium’s wholly owned subsidiary. As Cavium’s Motions to Intervene filed in district court makes it clear, Cavium itself is not a supplier of Dell. Ex.0009.005. Instead, it is Cavium’s wholly subsidiary QLogic that is Dell’s supplier. *Id* (“Cavium seeks to intervene because some of Alacritech’s infringement allegations in the Complaint in this action specifically accuse some Dell products of infringement based in part on their use of QLogic network adapters, including, among others, QLogic 57840S-k quad port 10GbE blade KR NDC”). There is no direct indemnification relationship between Cavium and Dell. The indemnification relationship is between QLogic and Dell. Cavium was involved in this lawsuit only because QLogic is Cavium’s

QLogic. Ex. 2009 at 1.

wholly owned subsidiary. Ex.2009.005 (“Cavium’s interest in this litigation is based on the involvement of its wholly-owned subsidiary, QLogic Corporation (‘QLogic’).”).

Cavium admitted that it “has a substantial, direct financial interest in the outcome of this litigation.” Ex.2009.005. Cavium also admitted that it “can offer technical knowledge and expertise that its customer lacks, as the designer and manufacturer of the QLogic technology targeted by Alacritech’s infringement allegations. Moreover, issues litigated in this action may affect other Cavium customers in the event that they are sued by Alacritech or its successors.”

Ex. 2009.005. The intimate relationship between Cavium and QLogic creates privity and makes QLogic a real party in interests.

B. Cavium Effectively Controls QLogic

Cavium has direct control over QLogic, not only because QLogic is Cavium’s wholly owned subsidiary, but also Cavium has made important decisions on behalf of QLogic in this case.

For example, in Cavium’s motion to intervene, it laid out the history of the patent infringement case. None of it ever mentions what QLogic did actively. Quite opposite, QLogic only took a passive role during the litigation in receiving an indemnification letter from Dell. Ex. 2009.005-.006. Since the beginning of this litigation, it appears that all communications have been conducted between

Cavium and Dell directly. *See* Ex. 2009.007 (“Cavium contacted Alacritech in early December . . . based on Cavium’s unavailability during the holidays . . . Cavium’s potential intervention in the action has been discussed with Alacritech’s counsel . . . Cavium attempted to avoid contested motion practice”).

Most importantly, Cavium made a key decision on behalf of QLogic, which is to “propose as an alternative that the parties could avoid motion practice by Alacritech dismissing the claims against Dell as to the QLogic products and *then filing a new complaint to assert those claims directly against QLogic.*” Ex. 2009.003. In other words, Cavium has the power to waive QLogic’s objection and offer it as a defendant as an exchange of dismissing Dell. The evidence of control here is overwhelming. In IPR2014-01380, the Board found the petitioner General Electric Company (“GE”) failed to disclose a real party interest, Oklahoma Gas & Electric Company (“OG&E”), whom the petitioner had control over in a district court case. *Id.* Here, for the same reason, Cavium also failed to disclose a real party in interest, its subsidiary and direct supplier of Dell, QLogic, over which Cavium fully controls.

C. Cavium Effectively Controls Dell

“There are multiple factors relevant to the question of whether a non-party may be recognized as a ‘real party in interest’ or ‘privy.’ A common consideration is whether the non-party exercised or could have exercised control over a party’s

participation in a proceeding.” Office Patent Trial Practice Guide, 77 Fed. Reg. at 48759-60. “Absolute control, however, is not necessary. Instead, a nonparty will be found to have control if it ‘has the actual measure of control or opportunity to control that might reasonably be expected between two formal coparties.’ Moreover, actual control is not required; the opportunity to exert the appropriate level of control is sufficient.” IPR2014-01380, Paper No. 34 at 8 (April 5, 2015) (internal citations omitted).

In IPR2014-01380, the petitioner GE’s control over the district court defendant was based on their agreement stating that “Petitioner would ‘provide a full and unqualified defense to OG&E.’” *Id.*¹⁰ Cavium has a similar agreement with Dell. Cavium’s litigation in-house declares in the district court case that “Dell has requested that Cavium indemnify, defend and hold it harmless from Alacritech, Inc.’s (“Alacritech”) claims in this action that concern QLogic

¹⁰ In IPR2014-01380, the petitioner GE argued that “only privity with respect to the present *inter partes* review proceeding, not privity with respect to the [district court] lawsuit, is relevant to the application of the § 315(b) bar,” but this was expressly rejected by the Board. *See* IPR2014-01380, Paper No. 34 at 11 (“We do not agree with Petitioner’s reading of § 315.”). Therefore, privity can be found based on Cavium’s control over Dell.

products, based on an indemnity agreement relating to Dell's purchases of QLogic products. . . . Based on the nature of the claims at issue in this suit, *Cavium has agreed to defend and partially indemnify Dell* as to the claims against Dell products incorporating Cavium/QLogic's technology ." Ex. 2003 at 1 (emphasis added). Indemnification is a factor favoring finding of privity. *See* IPR2014-01380, Paper No. 34 at 7-8 ("indemnitors . . . assume control of litigation against the parties they indemnify."); *see also* *Benson & Ford, Inc. v. Wanda Petroleum Co.*, 833 F.2d 1172, 1174 (5th Cir. 1987). Just as in IPR2014-01380, Cavium had control over Dell through the indemnification agreement but failed to disclose Dell as a real party in interests.

D. Dell Desires Review of the '948 Patent

A "real party in interest" is "the party that desires review of the patent." *See* Office Patent Trial Practice Guide, 77 Fed. Reg. at 48759; *see also* IPR2014-00488, Paper No. 52 at 9-10 (March 16, 2015). Here, Dell is the party originally accused of infringing the '948 patent in the district court case, not Cavium. Thus, Dell has an interest in the claims of the patents being determined to be unpatentable, which would allow it to avoid liability in the district court case. Cavium's products were not accused in the original pleading. Even if Dell has an indemnification agreement with its supplier, such supplier is not Cavium but QLogic.

Thus, it is Dell, not Cavium, that desires review of the '948 Patent, which further suggests privity exists between Dell and Cavium.

E. Intel Has Effective Choice as to the Legal Theories and Proofs of Dell and Cavium

Intel, another intervenor in the district court case, has effective choice of invalidity theories and proofs, as shown by the activities of Cavium. Just like Cavium, Intel is also a supplier of Dell. *See* Ex. 2002.006. After Intel filed this Petition, Cavium also filed an almost verbatim petition. *Compare* IPR2017-01729 with IPR2017-01395. Moreover, the petitions filed by Intel and Cavium also share an identical declaration from the same expert, Dr. Robert Horst. *Compare* IPR2017-01729, Ex. 1003 with IPR2017-01395, Ex. 1003. In Intel's Petition, Dr. Horst declares that he "has been retained on behalf of Petitioner Intel Corporation" (IPR2017-01395, Ex. 1003 at 004), and in Cavium's petition Dr. Horst declares that he "has been retained on behalf of Petitioner Cavium, Inc." (IPR2017-01729, Ex. 1003 at 004).

"A nonparty will be found to have control if the nonparty has 'effective choice as to the legal theories and proofs to be advanced in behalf of the party to the action,' as well as 'control over the opportunity to obtain review.'" IPR2014-01380, Paper 34 at 8 (April 5, 2015). The fact that Cavium chose to adopt not only Intel's invalidity theories but use identical language and the same expert is strong

evidence that these IPR petitions were orchestrated among the petitioners and Intel had effective choice as to the legal theories and proofs to be advanced. The two almost verbatim petitions filed by two intervenors associated with Dell is strong evidence that Dell, Intel, and Cavium are in privity and cooperating not only in the district court case but also in the IPR proceedings.

F. Finding QLogic, Dell, and Intel Are Real Parties in Interest Is Consistent with Legislative Intent

Finding that QLogic, Dell, and Intel are real parties in interest is also consistent with the express legislative intent concerning the need for quiet title. *See* 157 Cong. Rec. S1034, S1041 (Mar. 1, 2011) (Sen. Kyl) (stating “the present bill does coordinate *inter partes* . . . review with litigation . . . setting a time limit for seeking . . . review if the petitioner . . . is sued for infringement of the patent”). Congress “recognizes the importance of quiet title to patent owners to ensure continued investment resources.” H.R. Rept. No 112-98, at 48 (2011) (Judiciary Committee Report on H.R. 1249, June 1, 2011). Changes to the statutory structure “are not to be used as tools for harassment or a means to prevent market entry through repeated litigation and administrative attacks Doing so would frustrate the purpose of the section as providing quick and cost effective alternatives to litigation.” *Id.* “It would divert resources from the research and development of inventions.” *Id.* If Cavium were to be able to institute this IPR

without adding QLogic, Dell or Intel as real parties in interest, QLogic, Dell, and Intel would be able to use the same prior art relied upon here in other cases even if the patent is held to be valid in this IPR. This would be harassment through repeated litigation in violation of legislative intent, as QLogic, Intel, and Dell would be able to “double dip” and use the same invalidity theories to defend the same accused products twice. Indeed, Intel has already filed almost identical IPR petitions.

In view of the above, under the totality of circumstances Cavium is in privity with QLogic, Dell and Intel; thus, QLogic, Dell, and Intel are unnamed real parties in interest to this Petition. Office Patent Trial Practice Guide, 77 Fed. Reg. at 48759-60. Alacritech respectfully requests the Board to deny instituting this IPR because it fails to disclose QLogic, Dell, and/or Intel.

At the very least Alacritech is entitled to seek early discovery on this issue.

**X. ALACRITECH RESERVES ITS RIGHTS UNDER THE PENDING
OIL STATES CASE AT THE UNITED STATES SUPREME COURT**

As the Board is aware, the United States Supreme Court is currently considering certain aspects of the *inter partes* review process in *Oil States Energy Servs. LLC v. Greene’s Energy Group, LLC*, Case No. 16-712, *certiorari granted* (U.S. Jun. 12, 2017). To the extent the Supreme Court rules in *Oil States* before the Decision on Institution in this matter, Alacritech respectfully reserves its right

to timely request permission to brief the implications of that decision on this proceeding.

XI. THE BOARD SHOULD DECLINE INSTITUTION UNDER 35 U.S.C. § 325(D) BECAUSE THE PETITION RELIES ON PRIOR ART ALREADY CONSIDERED BY THE OFFICE

As noted above, Thia and Tanenbaum were already considered by the Patent Office during the original prosecution of the '948 Patent. Further, Stevens is relied upon to teach header prediction, which was already disclosed in Tanenbaum; thus, Stevens is cumulative to Tanenbaum.

In IPR2016-01309, the Board denied institution of a petition that “presents *substantially the same art* or arguments as those previously presented to the Office.” Paper 11, Decision Denying Institution at 2 (emphasis added). The Board noted that “[i]nstitution of *inter partes* review is discretionary. Our discretion on whether to institute is guided by 35 U.S.C. § 325(d), which states that ‘the Director may take into account whether, and reject the petition or request because, the same or substantially the same prior art or arguments previously were presented to the Office.’” *Id.* at 6. In that petition, like the current Petition, “Petitioner is thus asking the Board, essentially, to second-guess the Office’s previous decision on substantially the same issues.” *Id.* at 12. The Board also acknowledged in that case “the burden and expense to Patent Owner in having to defend [the challenged patent] based on substantially the same prior art or arguments already considered

by the Office.” *Id.* This form of second-guessing the original Examiner, according to the Board, was not “an efficient use of Board or party resources.” *Id.*¹¹

As in IPR2016-01309, the Board should deny institution of this Petition because it only presents prior art already considered by the Patent Office and no new arguments.

XII. CONCLUSION

For the foregoing reasons, there is no reasonable likelihood of Petitioner prevailing with respect to even one of the challenged claims. Accordingly, the Petition should be denied pursuant to 35 U.S.C. § 314(a).

¹¹ See also *Cultec, Inc. v. Stormtech LLC*, IPR2017-00777, Paper 7 at 13 (declining to review because “the same or substantially the same prior art or arguments as are presented in the Petition previously were presented to the Office.”); *Unified Patents Inc. v. John L. Berman*, IPR2016-01571, Paper 12 at 2-4.

Date: November 2, 2017

Respectfully submitted,

By: /James M. Glass/

James M. Glass (Reg. No. 46,729)

QUINN EMANUEL URQUHART &

SULLIVAN, LLP

51 Madison Ave., 22nd Fl.

New York, NY 10010

Tel: 212-849-7000

Fax: 212-849-7100

Email: jimglass@quinnemanuel.com

*Attorney for Patent Owner –
Alacritech*

CERTIFICATION UNDER 37 C.F.R. § 42.24

Under the provisions of 37 C.F.R. § 42.24, the undersigned hereby certifies that the word count for PATENT OWNER'S PRELIMINARY RESPONSE totals 10,717 words, not including the table of contents, table of authorities, mandatory notices under § 42.8, a certificate of service or word count, or appendix of exhibits or claim listing words, which is less than the 14,000 allowed under 37 C.F.R. § 42.24(a)(i).

Date: November 2, 2017

/s/ James M. Glass

James M. Glass (Reg. No. 46,729)

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), the undersigned hereby certify that
PATENT OWNER'S PRELIMINARY RESPONSE, and all exhibits and other
documents filed together with the preliminary response, were served on November
2, 2017 by filing these documents through the Patent Review Processing System,
as well as by e-mailing copies to:

Patrick D. McPherson,
Reg. No. 46,255
505 9th Street N.W., Suite 1000
Washington, DC 20004
P: (202) 776-7800
F: (202) 776-7801
PDMcPherson@duanemorris.com

Date: November 2, 2017

Respectfully submitted,

By: /James M. Glass/
James M. Glass (Reg. No. 46,729)
QUINN EMANUEL URQUHART &
SULLIVAN, LLP
51 Madison Ave., 22nd Fl.
New York, NY 10010
Tel: 212-849-7000
Fax: 212-849-7100
Email: jimglass@quinnemanuel.com

*Attorney for Patent Owner –
Alacritech*