UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD. and
SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioner,

v.

RADIAN MEMORY SYSTEMS, LLC,
Patent Owner.

Case IPR2025-01350
Patent 11,740,801

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I. INTRODUCTION 1

II. PETITIONER'S SINCLAIR COMBINATIONS FAIL (GROUNDS I, III, IV) 2

A. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success. 2

1. Petitioner Does Not Say What Its Combination Of Sinclair And Yamada Actually Is. 3

2. Petitioner Combines Mutually Inconsistent Embodiments Of Sinclair Without Explanation. 9

B. Petitioner Does Not Show That Its Sinclair Combinations Teach "a maintenance request issued by the host." 17

1. Petitioner Does Not Show Any "maintenance request issued by a host" In Sinclair. 18

2. Petitioner Does Not Show That Yamada Teaches Any "maintenance request issued by the host" In Its Combination. 23

C. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller Having Logic To "identify a specific one of the respective subdivisions … ." 24

D. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller "provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value." 27

III. PETITIONER'S TERASAKI GROUND FAILS (GROUND II). 37

A. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success. 37

1. Petitioner Does Not Say What Its Combination Of Terasaki And Yamada Actually Is. 37

2. Petitioner's Alleged Motivation To Modify Terasaki Is Conclusory. 40

B. Petitioner Does Not Show That Its Terasaki Combination Teaches Any "maintenance request issued by the host." 43

1. Petitioner Does Not Show Any "maintenance request issued by [a] host" In Terasaki. 43

2. Petitioner Does Not Show That Yamada Teaches Any "maintenance request issued by the host" In Its Combination. 51

C. Petitioner Does Not Show That Its Terasaki Combination Teaches Logic To Cause A Memory Controller To "identify a specific one of the respective subdivisions . . . ." 53

D. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller "stor[ing] a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host." 56

E. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller "provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value." 61

IV. CONCLUSION 67

TABLE OF AUTHORITIES

Page(s)

COURT DECISIONS

Arendi S.A.R.L. v. Apple Inc.,
832 F.3d 1355 (Fed. Cir. 2016) 40

Graham v. John Deere Co.,
383 U.S. 1 (1966).....8

In re Kotzab,
217 F.3d 1365 (Fed. Cir. 2000) 9

In re Magnum Oil Tools Int'l, Ltd.,
829 F.3d 1364 (Fed. Cir. 2016) 24, 36, 53

KSR Int'l Co. v. Teleflex Inc.,
550 U.S. 398 (2007).....8

LBT IP I LLC v. Apple Inc.,
2023 U.S. App. LEXIS 14376 (Fed. Cir. 2023) 22

Metalcraft of Mayville, Inc. v. Toro Co.,
848 F.3d 1358 (Fed. Cir. 2017) 17

Microsoft Corp. v. Biscotti, Inc.,
878 F.3d 1052 (Fed. Cir. 2017) 22

Microsoft Corp. v. FG SRC, LLC,
860 F. App'x. 708 (Fed. Cir. 2021) 60

Netflix, Inc. v. DivX, LLC,
84 F.4th 1371 (Fed. Cir. 2023) 61, 65

Novartis AG v. Torrent Pharms. Ltd.,
853 F.3d 1316 (Fed. Cir. 2017) 41

PersonalWeb Techs. LLC v. Apple, Inc.,
848 F.3d 987 (Fed. Cir. 2017) 2

Sisvel Int'l. S.A. v. Sierra Wireless, Inc.,
82 F.4th 1355 (Fed. Cir. 2023) 40, 66

Trintec Indus., Inc. v. Top-U.S.A. Corp.,
295 F.3d 1292 (Fed. Cir. 2002) 22

Virtek Vision Int'l ULC v. Assembly Guidance Sys., Inc.,
97 F.4th 882 (Fed. Cir. 2024) 17

AGENCY DECISIONS

Matterport, Inc. v. Appliance Computing III, Inc.,
IPR2021-00939, Paper 15 (Nov. 23, 2021) 8

UiPath, Inc. v. Rule 14 LLC,
IPR2024-01304, Paper 8 (Mar. 14, 2025) 42

Valeo N. Am., Inc. v. Magna Elecs., Inc.,
IPR2014-01206, Paper 13 (Dec. 23, 2014) 8

Xerox Corp. v. Bytemark, Inc.,
IPR2022-00624, Paper 9 (Aug. 24, 2022) (precedential)..... 21, 50

STATUTES

35 U.S.C. § 103 8

35 U.S.C. § 312(a)(3).....57

REGULATIONS

37 C.F.R. § 42.104(b)(4).....57

37 C.F.R. § 42.6(a)(3) 41

EXHIBIT LIST	
2001	Lowenstein Declaration In Support of Notice of Intent
2002	Woo Declaration In Support of Notice of Intent
2003	Courtney Declaration In Support of Notice of Intent
2004	Zhou Declaration In Support of Notice of Intent
2005	Declaration of Mike Jadon, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. May 6, 2025), Dkt. 43-13 [Jadon-Decl.]
2006	Laura Caulfield, *Project Denali to define flexible SSDs for cloud-scale applications*, MICROSOFT AZURE BLOG (Mar. 20, 2018), https://azure.microsoft.com/en-us/blog/project-denali-to-define-flexible-ssds-for-cloud-scale-applications/ [Project-Denali-Article].
2007	Defendants Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc.'s Amended Answer, Defenses and Counterclaims, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. Sept. 8, 2025), Dkt. 67 [Amended-Answer]
2008	Memorandum re: Interim Processes for PTAB Workload Management (Mar. 26, 2025) [3/26/25 Memo]
2009	Plaintiff's Complaint for Patent Infringement, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. Dec. 24, 2024) Dkt. 1 [Complaint]
2010	Order On Plaintiff's Motion to Withdraw Its Motion For Preliminary Injunction and Vacate The Hearing Set Before The Court, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. July 11, 2025), Dkt. 64 [Order]
2011	Plaintiff's Opposed Motion to Dismiss, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. July 17, 2025), Dkt. 69 [Motion-to-Dismiss]

2012	First Amended Docket Control Order, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. Apr. 14, 2025), Dkt. 32 [Docket-Control-Order]
2013	Samsung's P.R. 3-3 And 3-4 Invalidity Contentions, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. June 18, 2025) [Invalidity-Contentions]
2014	Statement of Interest of the United States of America *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. June 24, 2025) Dkt. 52 [US Statement of Interest]
2015	Press Release, TrendForce, NAND Flash Revenue Surged Over 20% in 2Q25, SK Group Market Share Jumped to 21%, Says TrendForce (Aug. 28, 2025), https://www.trendforce.com/presscenter/news/20250828-12688.html [TrendForce]
2016	[How much do you know about flash memory? - Part 4] The World's No. 1 Flash Memory Company, Samsung Electronics, and Their Unending Journey, Samsung Tech Blog (June 29, 2020), https://semiconductor.samsung.com/news-events/tech-blog/all-there-is-to-know-about-flash-memory-part4-samsung-electronics-world-no1-in-flash-memory-still-learning-still-innovating/ [Samsung-Article]
2017	Samsung's Opposition to Radian's Motion to Dismiss, *Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.*, 2-24-cv-01073 (E.D. Tex. Oct. 3, 2025) Dkt. 76 [Opposition]
2018	Median and Average Time To Trial for Judge Rodney Gilstrap, E.D. Tex., Docket Navigator [Gilstrap Time to Trial]
2019	U.S. Publication No. 2009/0271562 A1 ("Sinclair")
2020	Declaration of Mark Jones, Ph.D. [Jones-Decl.]

All emphases in quotations are added unless otherwise stated.

I. INTRODUCTION

The Petition should be denied because it fails to demonstrate a reasonable likelihood of prevailing on any challenged claim. The Petition fails to meet this threshold for multiple independently sufficient reasons across all asserted grounds.

Petitioner never articulates coherent combinations for either set of grounds. Petitioner's combinations all involve Yamada, but Petitioner never quite says how: Petitioner simply mentions Yamada in connection with the primary references, labels various portions of the cited art with claim language, and asks the Board to infer that Yamada combinations would have properties that Petitioner's base references do not. *See* Sections II.A.1, III.A.1 *infra*. Petitioner's proposed Sinclair grounds also combine incompatible embodiments of Sinclair without explanation, *see* Section II.A.2 *infra*, while the description of Petitioner's proposed Terasaki combination effectively amounts to a single sentence. *See* Section III.A.2 *infra*. Without a clear account of its contemplated combinations, Petitioner cannot show a motivation to combine, let alone a reasonable expectation of success.

Given these defects in articulating its combinations, it is no surprise that Petitioner also fails to show that several claim limitations are taught in its grounds. *See* Sections II.B-II.D, III.B-III.E, *infra.* The Board should deny institution.

II. PETITIONER'S SINCLAIR COMBINATIONS FAIL (GROUNDS I, III, IV).

Petitioner's Grounds I, III, and IV rely on U.S. Publication No. 2009/0271562 A1 ("Sinclair"), *see, e.g.*, Pet. iii—a publication Petitioner identifies as "Exhibit 1004," although Exhibit 1004 is an unrelated patent involving the same named inventor. *See id.*; *see also* Ex. 1004 (U.S. Patent No. 7,739,444). Because Petitioner did not elsewhere supply Sinclair to the Board in this proceeding, Patent Owner provides Sinclair as Exhibit 2019.

A. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.

Petitioner's Sinclair grounds should be rejected because Petitioner does not adequately explain how its references would be combined. As the Federal Circuit has made clear, an obviousness finding cannot stand without a "clear[] expla[nation]" of "*how* the combination of the two references was supposed to work." *PersonalWeb Techs. LLC v. Apple, Inc.*, 848 F.3d 987, 994 (Fed. Cir. 2017) (emphasis in original). Indeed, "a clear, evidence-supported account of the contemplated workings of the combination ***is a prerequisite*** to adequately explaining and supporting a conclusion that a relevant skilled artisan would have been motivated to make the combination and reasonably expect success in doing so." *Id*. Petitioner fails to provide such an account here. *See* Section II.A.1 *infra*. Petitioner's combinations should also be rejected because they combine

incompatible embodiments of Sinclair without analysis of why or how these disparate features would be present in a single system. *See* Section II.A.2 *infra.*

1. Petitioner Does Not Say What Its Combination Of Sinclair And Yamada Actually Is.

All of Petitioner's Sinclair grounds depend upon Petitioner's Sinclair-Yamada Ground I challenge to claim 1,[1] where Petitioner does not discuss its combination outside of argument about a memory controller storing a value relating to a "maintenance request issued by [a] host." *See generally* Pet. 29-30. At this point, Petitioner has identified Sinclair's memory controller as the claimed "memory controller," Pet. 13-14,[2] the initiation of a "flush operation" in Sinclair as the claimed

[1] The '801 patent has only one independent claim, and Petitioner's Sinclair-based arguments regarding dependent claims incorporate Petitioner's Ground I challenge to claim 1. *See, e.g.*, Pet., 38-65 (Ground I challenges to dependent claims), 92 (Ground III), 94 (Ground IV).

[2] Sinclair uses the terms "controller," "system controller," "memory system controller," and "memory controller" interchangeably. *See, e.g.*, Ex. 2019 [Sinclair] Fig. 1 (featuring a "memory controller" 108), Fig. 2 (with component labeled 108), [0009] (referring to Figure 2's "flash memory system controller," [0044] (referring

"maintenance request," *see, e.g.*, Pet. 25,[3] and Sinclair's host as the claimed "host," *see* Pet. 27, and Petitioner is trying to argue that Sinclair's host issues (or obviously could issue) what it has identified as a "maintenance request." Petitioner implicitly introduces its combination of Sinclair with Yamada as follows:

> Yamada discloses moving maintenance operations from the storage controller to the host. EX1005, ¶10 (desirable to perform "maintenance" on the memory controller so that "device conditions" may be considered). For instance, Yamada discloses that a maintenance determination unit in the access module 130 may include a "command generation unit" 154 that "generated a maintenance command for performing maintenance for the address of the nonvolatile memory device selected by the maintenance determination unit" 152[.] *Id.*, ¶22; EX1003, ¶117. Figure 1 illustrates the access module [host] and its "maintenance determination unit" and "command generation unit" that "generates a maintenance command for causing maintenance for data retention, such as refresh, to be performed for an address stored in the maintenance entry list 153":

---

to "the controller 108" of Figure 1), [0045] (referring to the "system controller 108" of Figure 2)

[3] Petitioner uses the term "***maintenance*** flush operation," *see* Pet. 27-28, presumably to bolster its identification with the claimed "maintenance request." Sinclair uses the term "flush operation."

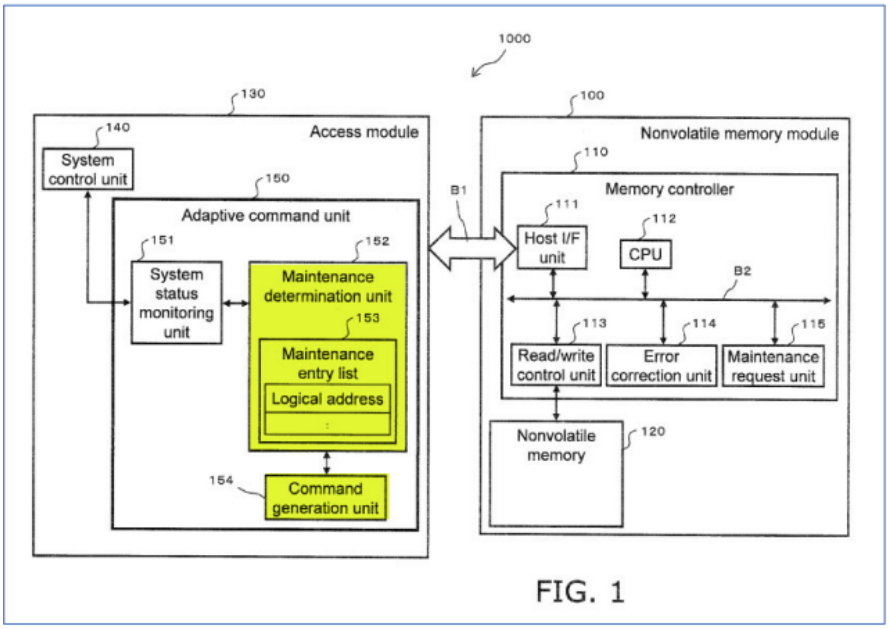


FIG. 1

EX1005, Fig. 1 (annotated), ¶¶89, 90 ("The maintenance command includes a logical address of a block that needs maintenance (hereafter referred to as a "maintenance-needed block") stored in the maintenance entry list 153."); EX1003, ¶117. In the combined Sinclair-Yamada system, the host would issue the flush operation [maintenance request issued by the host] for the pink blocks that need to be relocated [physically erased].EX1003, ¶118.

Pet. 28-30 (bracketed insertions in original).

To state that "[i]n the combined Sinclair-Yamada system, the host would issue the flush operation" does not identify what "the combined Sinclair-Yamada system" is. In particular, while Petitioner makes passing mention of Yamada components ("maintenance determination unit," "command generation unit," "maintenance entry list"), and while Petitioner includes a highlighted figure showing these components within Yamada's "access module," Petitioner has not explained how these

components operate in Yamada beyond what it has said above,[4] let alone stated how or whether these components are present in its combination.

Yamada's components are parts of an integrated system. Yamada discloses, for example, that its "maintenance determination unit" operates on input from a component in Yamada that Petitioner does not mention—a "maintenance request unit" within Yamada's memory controller (on the right hand side of Yamada's Figure 1), which populates the "maintenance entry list" that Petitioner mentions. *See, e.g.*, Ex. 1005 [Yamada] [0089] ("The maintenance determination unit 152 stores a logical address included in ***each maintenance request transmitted from the maintenance request unit*** 115 in a maintenance entry list 153"); [0100]-[0106] (describing a "Data Maintenance Command Generation Cycle" that begins with input from Yamada's "maintenance request unit"); Ex. 2020 [Jones-Decl.] ¶ 34.

Whatever the details of operation of Yamada's components, it is unclear if Petitioner proposes that Yamada's "maintenance determination unit" and "command generation unit" are ***part*** of an alleged "host" in "the combined Sinclair-Yamada system," or if Petitioner is only arguing that Yamada somehow ***teaches*** via these components that a POSITA would modify Sinclair so that Sinclair's modified "host"

[4] The only previous mention of Yamada in the Petition is a sentence alleging that Yamada is prior art. Pet., 12.

issues what Petitioner identifies as "maintenance requests."

In the former case—where a copy of Yamada's "maintenance determination unit" exists within the "host" of the combined system—Petitioner leaves open how Yamada's "maintenance determination unit" receives information about what alleged "maintenance requests" to issue. Would it do this ***as it does in Yamada***—that is, from a copy of Yamada's "maintenance request unit" within the "controller" of the combined system? Petitioner, who does not discuss Yamada's "maintenance request unit" at all, does not say. *See* Pet. 29. In focusing entirely on the ***host*** of its "combined Sinclair-Yamada system"—and telling the Board only that "the host would issue [a] flush operation," Pet. 30—Petitioner does not mention any modification to Sinclair's ***controller*** as part of its "combined Sinclair-Yamada system." *See* Pet. 29-30 (discussing host-side aspects of Yamada). Is Sinclair's controller being used without modification? Again, Petitioner does not say.

If Petitioner proposes to modify Sinclair in a way inspired by Yamada that does ***not*** incorporate Yamada's components, Petitioner does not identify what component in Sinclair's host issues alleged "maintenance requests," how that component gets information about which alleged "maintenance requests" to issue, or on what basis the host issues such requests. Elsewhere, Petitioner argues that alleged "maintenance requests" in Sinclair ***already*** come from Sinclair's host, *see* Pet. 27-28 (referring to what Petitioner terms the "host-issued flush operation

embodiment")—raising the question of how Petitioner's Sinclair-Yamada combination differs ***at all*** from unmodified Sinclair.

Petitioner's statement that "the host would issue the flush operation" in what Petitioner terms "the combined Sinclair-Yamada system," Pet. 30, is at most an argument ***about*** a system; it does not identify the system. Petitioner's refusal to be clear about how it proposes to modify Sinclair is contrary to the law of obviousness, which requires identification of "the differences between the claimed invention and the prior art …." 35 U.S.C. § 103; *see also, e.g.*, *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007) (citing *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966)); *see also, e.g.*, *Valeo N. Am., Inc. v. Magna Elecs., Inc.*, IPR2014-01206, Paper 13, 15 (Dec. 23, 2014) (denying institution) ("Petitioner's obviousness analysis is incomplete in that it: fails to state explicitly the differences between the [prior art] and the claimed subject matter, does not explain persuasively the relevance of the evidence, and presents conclusory assertions regarding how the references are combined."); *Matterport, Inc. v. Appliance Computing III, Inc.*, IPR2021-00939, Paper 15, 16-17 (Nov. 23, 2021) (denying institution when "Petitioner does not identify differences between [cited art] and element 1.2 or how Petitioner proposes to combine or modify the asserted prior art").

2. Petitioner Combines Mutually Inconsistent Embodiments Of Sinclair Without Explanation.

Even within a single reference, a motivation must be shown for combining features of different embodiments, or otherwise modifying the teachings of a reference. *See, e.g.*, *In re Kotzab*, 217 F.3d 1365, 1370 (Fed. Cir. 2000). In arguing that claim 1 is obvious, however, Petitioner does not adhere to a single embodiment of Sinclair, but instead makes piecemeal selections from mutually incompatible embodiments. Because Petitioner never acknowledges this aspect of its arguments, Petitioner does not articulate why a POSITA would have been motivated to make Petitioner's selections, let alone do so with a reasonable expectation of success.

Petitioner relies upon what Sinclair refers to as its "storage address re-mapping" (or "STAR") "functionality." Ex. 2019 [Sinclair] [0065]; *see generally, e.g.*, Pet. 25-31 (relying on a "flush operation" performed during Sinclair's "storage address re-mapping algorithm" for a claim limitation), 27 (referring to the "storage address re-mapping functionality" of Sinclair's paragraph 65 as "re-mapping (STAR) functionality"). Sinclair describes this functionality in two different types of systems: embodiments where "the storage address re-mapping (STAR) functionality" is "contained ***totally within the memory system***," and an embodiment (Sinclair's Figure 12) in which "the storage address re-mapping functionality is contained ***totally within the host***." Ex. 2019 [Sinclair] [0059]. Sinclair's Figures 10

and 11 depict examples of the first type of system, in which the "STAR" functionality is contained "totally within" a "memory system" (labeled 1004 in Figure 10, and 1102 in Figure 11),

Fig. 10

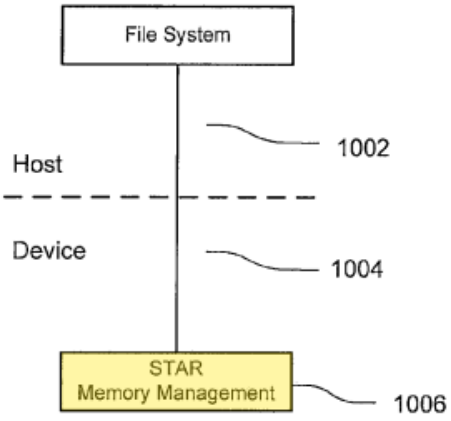
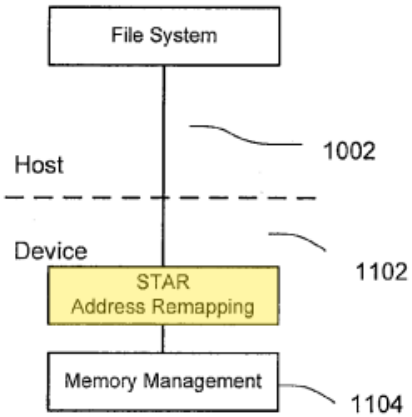
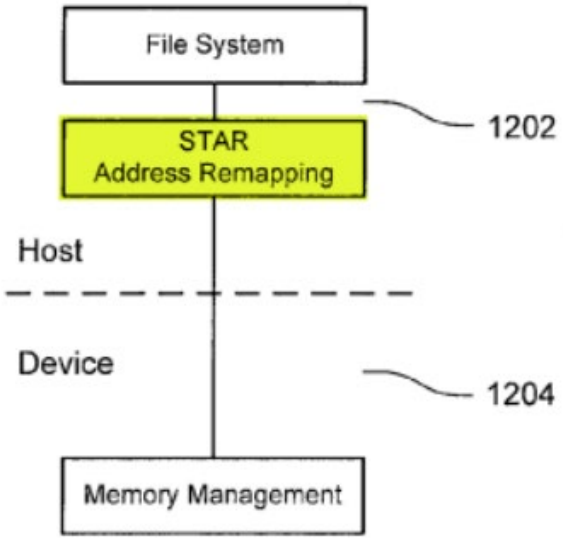


Fig. 11



Ex. 2019 [Sinclair] Figs. 10-11 (highlighted by Patent Owner), [0059], while Sinclair's Figure 12 is Sinclair's system in which the "STAR" functionality is contained "totally within" Sinclair's "host" 1202:

Fig. 12



Ex. 2019 [Sinclair] Fig. 12 (as highlighted by Petitioner, Pet. 28), [0059]; Ex. 2020 [Jones-Decl.] ¶ 39. Petitioner alternates between Sinclair's Figures 10-11 and Figure 12 embodiments without explanation.

As one example, claim 1 recites a "host" that Petitioner identifies with Sinclair's host, *see, e.g.*, Pet. 27, and in connection with claim 1's recitation of a "maintenance request issued by the host," Petitioner identifies the claimed "maintenance request" with "the storage address re-mapping algorithm … initiat[ing] a flush operation" in Sinclair, and expressly relies upon Figure 12's placement of Sinclair's STAR functionality to argue that this alleged "maintenance command" is "issued by the host." Pet. 27-28; *see also, e.g.*, Pet. 27 (expressly arguing that "[i]n some embodiments, 're-mapping (STAR) functionality' may be

contained 'within the host'"), 28 (including an annotated copy of Figure 12 and referring to it as "the host-issued flush operation embodiment" of Sinclair); *see also id.*, 30 (citing Sinclair's Figure 12 as alleged motivation "to combine the teachings of Sinclair and Yamada"), 31 (citing Sinclair's Figure 12 as alleged evidence of a reasonable expectation of success "in combining Sinclair and Yamada"). Petitioner does not argue that Sinclair's Figure 11 embodiment discloses this limitation—and a POSITA would understand that in Sinclair's Figure 11 embodiment, what Petitioner identifies as the claimed "maintenance request"—*i.e.*, the initiation of a "flush operation" by Sinclair's "storage address re-mapping algorithm"—occurs entirely within the controller and does not involve any request issued by Sinclair's host. *See, e.g.*, Ex. 2019 [Sinclair] [0077] (describing the beginning of the "block flush process" by stating that "[t]he storage address re-mapping algorithm, ***executed by the controller*** 108 independently for each bank 107A-107D ***in the implementation of FIG. 11***," designates a block to be flushed); Ex. 2020 [Jones-Decl.] ¶ 40.

As another example of Petitioner's reliance on Sinclair's Figure 12, in connection with claim 1's recitation of "provid[ing] to the host, responsive to a query received from the host … the stored value," Petitioner identifies the claimed "stored value" with a "list of pink blocks that are to be relocated," Pet. 32, and exclusively relies on the location of STAR functionality in Sinclair's Figure 12 embodiment—

which it terms "the host-issued flush operation embodiment"—to argue that this "stored value" is "provided to the host." Pet. 32; *see also id.* (arguing that "for ***the host*** to perform the flush operation, it ***must be provided with*** … the list of pink blocks that are to be relocated"). Again, Petitioner does not argue that Sinclair's Figure 11 discloses this limitation, and a POSITA would recognize that in Sinclair's Figure 11, what Petitioner identifies as the claimed "stored value" is not provided to Sinclair's host. Ex. 2020 [Jones-Decl.] ¶ 41.

Having repeatedly relied on Sinclair's Figure 12 for some limitations, Petitioner switches to Figure 11 for other aspects of claim 1.

As one example, claim 1 recites a "storage device" comprising "a ***memory controller*** having logic to cause the ***memory controller*** to … receive write requests ... and, with respect to each one of the write requests, ***identify a specific one of the respective subdivisions*** dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests." Petitioner identifies the claimed "memory controller" with Sinclair's memory controller, *see, e.g.*, Pet. 13-14, and identifies the claimed "identify[ing] a specific one of the … subdivisions" with "identifying a currently open write megablock" in Sinclair. Pet. 17-19. The process of identifying a currently open write megablock in Sinclair involves the conversion of "host LBA addresses" to "DLBA addresses." *Id.* In arguing that this process is performed by Sinclair's "controller" (as its identifications

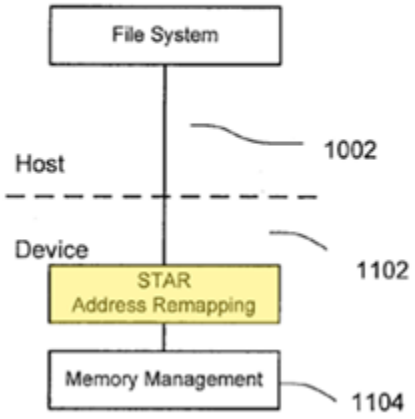
with elements of claim 1 would require), Petitioner expressly and exclusively relies upon disclosures from Sinclair's paragraphs 70-72, which describe the mapping of "host LBA addresses" to "DLBA addresses" in Sinclair's Figure 11 embodiment. *See, e.g.*, *id*., 18-20 (arguing that "Sinclair also discloses that its ***controller*** identifies the open write megablock," and extensively quoting from paragraphs 70-72). Sinclair's paragraph 70 expressly states:

> In this example, it is assumed that ***the system configuration of FIG. 11*** is being used, where the host LBA addresses are translated to an intermediate storage LBA address, also referred to as a DLBA address, ***in an application run by the controller 108*** in the memory system 102.

Ex. 2019 [Sinclair] [0070]; *see also, e.g.*, *id.* [0071] (referring to "the write algorithm managed by the controller 108"); Pet. 18-19 (expressly quoting the "in an application run by the controller" language of paragraph 70 while arguing that Sinclair's controller "identifies the open write megablock").

Sinclair's Figure 11 embodiment is an embodiment where "the storage address re-mapping (STAR) functionality" (which includes the mapping functions relied on by Petitioner) is contained "totally within" Sinclair's "memory system" 1102. Ex. 2019 [Sinclair] [0059].

Fig. 11



Ex. 2019 [Sinclair] Fig. 11 (highlighted by Patent Owner).

This difference from Sinclair's Figure 12 embodiment is ***exactly*** what Petitioner is relying upon to argue that the claimed "identify[ing] a specific one of the respective subdivisions" is performed by what Petitioner has identified as the claimed "controller." (Sinclair's "controller," of course, is part of Sinclair's "memory system." *See, e.g.*, Ex. 2019 [Sinclair] [0070] (specifically referring to "the ***controller*** 108 ***in the memory system*** 102" in connection with the Figure 11 embodiment), Fig. 1 (generally locating the "controller" within the "memory system" in embodiments of Sinclair); Ex. 2020 [Jones-Decl.] ¶ 45.) Sinclair expressly discusses the difference between Figures 11 and 12 in these terms, stating:

> The implementation of FIG. 12 would ***move*** the functionality of storage address re-mapping from the storage device 1204 ***to an application on the host 1202***. In this implementation, the function of mapping LBA addresses to DLBA addresses would be similar to that

> of FIG. 11, with ***the primary difference being that the translation would occur on the host*** 1202 and ***not in the memory device*** 1204.

Ex. 2019 [Sinclair] [0061]. (Here, Sinclair is using the term "memory device" synonymously with what it elsewhere calls a "memory system." *See, e.g.*, Ex. [0087] (referring to "the memory system 1204" in Figure 12); Ex. 2020 [Jones-Decl.] ¶ 45.)

With this "move [of] functionality" from Figure 11 to Figure 12, of course, the functionality that Petitioner uses to argue that Sinclair discloses "a ***memory controller*** with logic to cause the ***memory controller*** to identify a specific one of the respective subdivisions" would exist "not in the memory device," but "on the host." Because this is not where Petitioner's mapping onto claim 1 would require it to be, Petitioner relies on Sinclair's Figure 11 embodiment. But Petitioner does not address how its reliance on Figure 11 for one limitation is compatible with its reliance on Figure 12 for other limitations.

Sinclair itself does not discuss hybrid systems in which aspects of its "STAR" functionality are present in ***both*** its "host" and its "memory system." Ex. 2020 [Jones-Decl.] ¶ 46. Because Petitioner never acknowledges that it is effectively inventing such an embodiment by citing mutually incompatible embodiments across different claim limitations, it is impossible to know how Petitioner is proposing that a POSITA could reconcile the differences between the systems that Sinclair does

disclose, let alone do so with a reasonable expectation of success. Petitioner is effectively arguing only that "elements of the claimed invention [are] independently known in the prior art." *Metalcraft of Mayville, Inc. v. Toro Co.*, 848 F.3d 1358, 1367 (Fed. Cir. 2017). This is not enough to show a motivation to combine. *Virtek Vision Int'l ULC v. Assembly Guidance Sys., Inc.*, 97 F.4th 882, 887 (Fed. Cir. 2024).

B. Petitioner Does Not Show That Its Sinclair Combinations Teach "a maintenance request issued by the host."

Claim 1[5] recites a "memory controller" having logic relating to "fulfilling a ***maintenance request issued by the host***." Petitioner identifies Sinclair's memory controller as the claimed "memory controller," Pet. 13-14,[6] the alleged "initiat[ion]" of a "flush operation" in Sinclair as the claimed "maintenance request," *see, e.g.*, *id.*,

[5] *See supra* note 1.

[6] Sinclair uses the terms "controller," "system controller," and "memory system controller" interchangeably for its "memory controller." *See, e.g.*, Ex. 2019 [Sinclair] Fig. 1 (featuring a "memory controller" 108), Fig. 2 (depicting a component labeled 108), [0009] (referring to Figure 2 as a "flash memory system controller," [0044] (referring to "the controller 108" of Figure 1), [0045] (referring to the "system controller 108" of Figure 2).

25,[7] and Sinclair's host as the claimed "host." *See id.*, 27. But Petitioner does not show that Sinclair's host ever "issue[s]" what it identifies with the claimed "maintenance request," as the claim would require.

1. Petitioner Does Not Show Any "maintenance request issued by a host" In Sinclair.

Having made the identifications just mentioned—in particular, having vaguely associated an alleged "maintenance request" with a "flush operation [being] initiated," *id.*, 25—Petitioner clearly states that Sinclair's "flush operations" are themselves carried out at Sinclair's controller. Petitioner quotes Sinclair to the effect that "when the storage address re-mapping algorithm detects a need for more white blocks in the bank, the algorithm initiates a flush operation." *Id.*, 25 (quoting Ex. 2019 [Sinclair] [0065]). Petitioner states that during the flush operation, "***the memory controller*** determines and stores the number of pink blocks that need to be relocated." Pet. 25. Petitioner states that specific blocks to be relocated in the flush operation can be identified using the fact that "***the memory controller*** may maintain a list of pink blocks where the 'amount of valid data' is below a 'threshold,'" *id.*, 26, and that the relocation and erasure of those blocks completes the "flush operation."

[7] Petitioner uses the term "***maintenance*** flush operation," *see* Pet. 27-28, but Sinclair uses the term "flush operation." *See supra* note 3.

See, e.g., Pet. 25-26 (citing Ex. 2019 [Sinclair] Fig. 19). To repeat the essential point, whenever Petitioner bothers to identify the location of anything related to carrying out a "flush operation," Petitioner identifies that location as Sinclair's ***controller***, not Sinclair's host.

Petitioner argues, however, that Sinclair "discloses that the flush operation may be ***initiated*** by the host." Pet. 27. Here Petitioner begins to depart from the claim language. The claim, after all, refers to a controller fulfilling a "***maintenance request*** issued by the host"—and Petitioner's subsequent argument, which summarizes circumstances under which flush operations might take place, never identifies any purported "maintenance request" in Sinclair.

First, Petitioner argues that "Sinclair discloses that its flush operation ***performed in fulfilling a maintenance request*** may be initiated by a host activity, when a host 'flush cache' command occurs, or in response to host commands." *Id.* (internal citations omitted) (citing Ex. 2019 [Sinclair] [0112]). Here the language "performed in fulfilling a maintenance request" is Petitioner simply inserting the claim term "maintenance request" into an otherwise unexplained characterization of Sinclair. The cited portion of Sinclair does not identify any "maintenance request." It states:

> Preferably, the flush operation acts as a background operation, and thus functions only while the SSD or other portable flash memory

> device is idle so that host data write speeds are not affected. Alternatively, the flush operation may be utilized in a foreground operation that is active when the host is writing data. If ***flush operations are arranged as foreground operations***, ***these operations may be automatically suspended when host activity occurs or when a "flush cache" command signifies potential power-down*** of the SSD or portable flash memory device.

Ex. 2019 [Sinclair] [0112]. In saying that Sinclair's "flush operations" may be "automatically suspended when host activity occurs," or when a "flush cache" command occurs, Sinclair is plainly ***not*** saying that its flush operations are "initiated by a host activity," or "initiated … when a host 'flush cache' command occurs, or in response to host commands," as Petitioner would have it. *See* Pet. 27. Sinclair is saying that host activity and host "flush caches" in Sinclair may "***automatically suspend***[]" the operation of Sinclair's memory controller, such that "flush operations" that might otherwise have taken place at a given time do ***not*** take place. Ex. 2020 [Jones-Decl.] ¶ 52.

Petitioner then quotes the same disclosure in Sinclair to the effect that a "flush operation may be utilized in a foreground operation that is active when the host is writing data." Pet. 27 (quoting Ex. 2019 [Sinclair] [0112]). But a statement that a flush operation "may be utilized" in an operation "that is active when the host is writing data" is not a statement that Sinclair's host ***has requested*** that flush

operation—and does not, in any event, identify any candidate "maintenance request issued by the host" in Sinclair. Petitioner continues:

> A POSITA would understand these disclosures to teach that during foreground operation, the flush operation … is initiated by a "host activity" such as an activity which requires more white blocks to be created, or at minimum suggest that the maintenance request can be issued by the host.

Pet. 27. This ***still*** does not identify any "maintenance request issued by the host." And in view of the failure of Sinclair's own disclosure to provide any support for this assertion about how flush operations are "initiated," this statement about what "[a] POSITA would understand" rests entirely on a citation to Petitioner's expert, who only says the same thing. *See* Ex. 1003 [Litz-Decl.] ¶ 114. Such a statement should be given no weight. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9, 15 (Aug. 24, 2022) (precedential) (denying institution when expert's "cited declaration testimony is conclusory and unsupported, adds little to the conclusory assertion for which it is offered to support, and is entitled to little weight").

Still trying to find some way of locating a "maintenance request issued by [a] host" in Sinclair, Petitioner turns to Sinclair's Figure 12 embodiment. Quoting Sinclair, Petitioner observes that Figure 12 depicts an embodiment for which "'re-mapping (STAR) functionality' may be contained 'within the host.'" Pet. 27 (quoting Ex. 2019 [Sinclair] [0068]). Characterizing Sinclair's Figure 12 as "the ***host-issued***

flush operation embodiment"—again, characterizing Sinclair with claim language without explanation or supporting argument—Petitioner observes that Sinclair discloses that in embodiments of Sinclair's Figure 12, "the 'host [] may select pink blocks' for the flush operation." Pet. 28 (quoting Ex. 2019 [Sinclair] [0068]).

Petitioner then states that a POSITA would understand the host's potential involvement in the selection of "pink blocks" (which ultimately serve as ***inputs*** to Sinclair's flush operations, *see, e.g.*, Pet. 26) implies or at least "suggest[s] that" commands to perform flush operations themselves "can be issued by the host to the memory controller in various embodiments." *Id.*, 28. This assertion about what a POSITA supposedly would have understood is a thinly-veiled inherency argument. *Microsoft Corp. v. Biscotti, Inc.*, 878 F.3d 1052, 1065 (Fed. Cir. 2017) (statements that "a [POSITA] would understand the [evidence] as disclosing [the limitation]" could support an argument that a limitation was "inherently disclose[d] …"); *LBT IP I LLC v. Apple Inc.*, 2023 U.S. App. LEXIS 14376, *8 (Fed. Cir. 2023) (nonprecedential) (similar). To establish inherency, however, a Petitioner must demonstrate that what it identifies as the "management command issued by the host" is "necessarily present, not merely probably or possibly present, in the prior art." *Trintec Indus., Inc. v. Top-U.S.A. Corp.*, 295 F.3d 1292, 1295 (Fed. Cir. 2002) (cleaned up). Petitioner does not address the relevant standard, let alone show that it is satisfied. Indeed, in arguing that Sinclair's Figure 12 embodiment "suggest[s]

that the maintenance request ***can be*** issued by the host to the memory controller in ***various embodiments***," Pet. 28, Petitioner is effectively admitting that Sinclair's system does not ***require*** any putative maintenance request to be issued by its host—and if the situation is any different in unspecified "various embodiments" of Sinclair, Petitioner does not say how. Petitioner therefore fails to show the limitation.

2. Petitioner Does Not Show That Yamada Teaches Any "maintenance request issued by the host" In Its Combination.

In a final attempt to identify a "maintenance request issued by the host" in its combination, Petitioner turns to Yamada. Pet. 28-30 (arguing that "Yamada discloses moving maintenance operations from the storage controller to the host"). But Petitioner makes this identification in ***Yamada***, and does not relate this identification to its identification of other claim elements in Sinclair, or to elements of any combined system. Petitioner states:

> Yamada discloses that a maintenance determination unit in the access module 130 may include a "command generation unit" 154 that "generated a maintenance command for performing maintenance for the address of the nonvolatile memory device selected by the maintenance determination unit" 152.

Id., 29.

Here, Petitioner finally identifies a specific thing as a putative "maintenance request from a host": a "maintenance command" in Yamada, as generated by

Yamada's "command generation unit," within Yamada's "access module 130." But Petitioner does not connect this disclosure in Yamada to anything in Sinclair. Petitioner does not, for example, state that Sinclair's host would be modified as to contain a copy of Yamada's "command generation unit" in its combined system. *See id*. Nor does Petitioner explain how Sinclair's host would be modified as to ***implement the functionality*** of Yamada's "maintenance determination unit" and "command generation unit" if it does ***not*** contain these components. *See id.* Petitioner simply asserts that "[i]n the combined Sinclair-Yamada system, the host would issue the flush operation [**maintenance request issued by the host**] for the pink blocks that need to be relocated [**physically erased**]." *Id.*, 30 (bracketed insertion and boldface in original). This is conclusory, and does not show the limitation. *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016) ("To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements.").

C. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller Having Logic To "identify a specific one of the respective subdivisions … ."

Claim 1[8] recites a "storage device" comprising "a ***memory controller*** having logic to cause the ***memory controller*** to … receive write requests ... and, with respect

[8] *See supra* note 1.

to each one of the write requests, ***identify a specific one of the respective subdivisions*** dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests." Petitioner identifies the claimed "memory controller" with Sinclair's memory controller, *see, e.g.*, Pet. 13-14, the claimed "identify[ing] a specific one of the … subdivisions" with "identifying a currently open write megablock" in Sinclair, and the claimed "address accompanying the one of the write requests" with "host LBA addresses" provided by Sinclair's host. Pet. 18-19. Petitioner exclusively relies on Sinclair for this limitation. *See id.*

Petitioner does not show how Sinclair uses any host LBA addresses to identify the "currently open write megablock" in Sinclair—let alone that Sinclair performs any such identification "dependent on which one of the non-overlapping ranges encompasses" any host LBA address, as the claim would require under Petitioner's identifications.

As Petitioner recognizes, Sinclair's write algorithm maps incoming host data to "the currently open megapage in the currently open write megablock," and keeps writing there until the currently open write megablock is full. *See* Ex. 2019 [Sinclair] [0063]; *see also* Pet. 17-18 (quoting Sinclair's paragraph 63); Ex. 2020 [Jones-Decl.] ¶ 62. When a currently open write megablock is full, Sinclair states that "a new write block is allocated from a white block list." Ex. 2019 [Sinclair] [0064]. Sinclair

does not state that any LBA addresses provided by its host play any role in this process, and Petitioner does not explain how Sinclair otherwise teaches this aspect. As Dr. Jones explains:

> Petitioner asserts without support that the controller "identifies the open write megablock … dependent on which one of the intermediate storage LBA addresses … encompasses the host LBA addresses." Pet. 18. But the cited text shows no use of either a host LBA or a DLBA to select or otherwise identify which megablock is open for writing; Sinclair describes always writing to whatever megablock is already open. Petitioner's subsequent citation to Sinclair's Figures 16–17 and paragraph 70 shows only that during writing, the open write megablock is "associated with" an LBA run, and that Sinclair's storage address table ("SAT") maps LBA runs to DLBA runs while data is written until the megablock is fully programmed. This does not show the use of any LBA or DLBA in identifying a megablock to write to.

Ex. 2020 [Jones-Decl.] ¶ 64.

Petitioner's argument about this limitation also expressly relies upon Sinclair's paragraphs 70-72, which describe the mapping of "host LBA addresses" to "DLBA addresses" as they are performed in Sinclair's Figure 11 embodiment. *See, e.g.*, Pet. 18-20 (citing and quoting Ex. 2019 [Sinclair] [0070]-[0072]). As already noted, however, Petitioner elsewhere exclusively relies upon Sinclair's ***Figure 12*** embodiment for other aspects of claim 1, and Petitioner's arguments

regarding those aspects do not apply to Sinclair's Figure 11 embodiment. *See* Section II.A.2 *supra* (describing Petitioner's reliance on Sinclair's Figure 12 for "provid[ing] to the host, responsive to a query received from the host … the stored value"). Similarly, in Sinclair's Figure 12 embodiment, what Sinclair terms its "storage address re-mapping (STAR) functionality"—which includes the mapping of "host LBA" addresses to "DLBA addresses" relied upon here—is "contained ***totally within*** the host," and thus does not evidence a ***controller*** with logic to perform this functionality. *See* Ex. 2019 [Sinclair] [0059]; *see also id.*, [0061] (stating that "the primary difference" in "mapping LBA addresses to DLBA addresses" between Figure 12 and Figure 11 "being that the translation [in Figure 12] would occur on the host [] and not in the memory device"); Ex. 2020 [Jones-Decl.] ¶ 65. Petitioner's failure to adhere to a coherent narrative for how Sinclair applies to claim 1 is an additional reason why Petitioner fails to show a controller with logic to "identify a specific one of the respective subdivisions . . ." as claimed in its combined system.

D. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller "provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value."

Claim 1[9] recites, *inter alia*, a "storage device" comprising a "memory

[9] *See supra* note 1.

controller" having logic to cause the memory controller to "provide to the host, responsive to a ***query received from the host***, the information, as updated, and the stored value." Petitioner does not show how this limitation is taught in its Sinclair combinations.

Petitioner identifies Sinclair's memory controller with the claimed "memory controller," Sinclair's host with the claimed "host," "a list of red, pink and white blocks for a megablock and their valid page count" in Sinclair for the claimed "information, as updated," and "a listing of pink blocks that are to be relocated" in Sinclair for the claimed "value." Pet. 32.

To attempt to identify something in Sinclair as the claimed "query received from the host" that would prompt Sinclair's controller to provide Sinclair's host with what Petitioner identifies as the claimed "information, as updated" and "stored value," Petitioner exclusively relies upon Sinclair's Figure 12 embodiment. *See* Pet. 32; *see also* Pet. 27-28 (identifying Figure 12 as "the host-issued flush operation embodiment").

As already noted, however, Petitioner elsewhere exclusively relies upon Sinclair's Figure 11 embodiment for allegedly teaching other aspects of claim 1, and Petitioner's arguments regarding those aspects do not apply to Sinclair's Figure 12 embodiment. *See* Section II.A.2 *supra*. Similarly, Petitioner does not and cannot argue that Sinclair's Figure 11 teaches the claimed "query from the host"; Sinclair

expressly discloses, for example, that in its Figure 11 embodiment, "the storage address re-mapping (STAR) functionality"—which manages and uses the lists of blocks that Petitioner identifies with the claimed "information" and "stored value"—is "contained ***totally within*** the memory system." Ex. 2019 [Sinclair] [0059]. Sinclair says that locating this functionality "totally within" the memory system allows the system to "operate with a ***legacy host*** 1002 with ***no modifications required on the host*** 1002." Ex. 2019 [Sinclair] [0059]. A POSITA would understand from this and other disclosures in Sinclair that in the Figure 11 embodiment, there is no reason for the host to query the controller for the information identified by Petitioner, and no reason for the memory controller to be configured to provide that information to the host. Ex. 2020 [Jones-Decl.] ¶ 72. In failing to explain how Sinclair's distinct embodiments apply to claim 1, Petitioner thus fails to show how the claimed "query from the host" is present in its combined system.

As an alternative to relying on Sinclair for the claimed "query received from the host," Petitioner turns to Yamada. Petitioner argues that "Yamada discloses a maintenance determination unit with a maintenance entry list to perform maintenance operations" and that "Yamada teaches that the controller maintains information about the valid block count and that this information can be retrieved via host queries." Pet. 32-33 (citing Ex. 1005 [Yamada] [0032], [0096]).

The cited disclosures in Yamada do not show that Yamada's "information about the valid block count" is retrieved via host queries. The first paragraph cited by Petitioner refers to an "error correction unit" within Yamada's controller, and a "maintenance request unit" within Yamada's controller "generat[ing] a maintenance request." It reads in full:

> The error correction unit determines the number of errors included in data read from the nonvolatile memory, and corrects an error included in the read data. The maintenance request unit generates a maintenance request indicating that the nonvolatile memory needs maintenance to the access device when the number of errors determined by the error correction unit is greater than or equal to a criterion value.

Ex. 1005 [Yamada] [0032]. This does not refer to anything in Yamada being retrieved via host queries, but instead concerns operations internal to Yamada's controller.

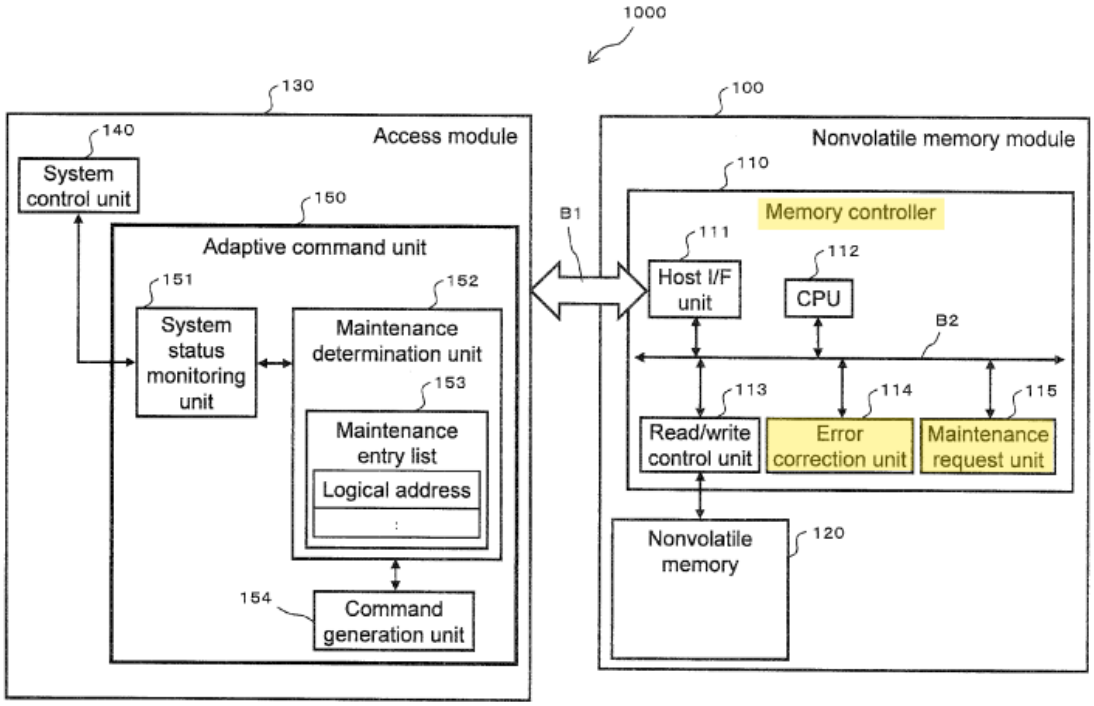


FIG. 1

See Ex. 1005 [Yamada], Fig. 1 ("Error correction unit" and "Maintenance request unit" within the "Memory controller" highlighted by Patent Owner).

The second paragraph cited by Petitioner, which also describes aspects of the "error correction unit" within Yamada's controller, says that a "CPU" on the controller "***transmits*** … data from the error correction unit" to an "access module" on Yamada's host:

> The error correction unit 114 determines whether the read data has an error, and determines the number of errors in the data. When detecting an error in the data, the error correction unit 114 corrects the erroneous data. After completing the error correction, ***the CPU 112***

> ***transmits the data from the error correction unit*** 114 ***to the access module*** 130, and completes the data reading process (S201).

Ex. 1005 [Yamada] [0096]. This disclosure does not show any queries issued by what Petitioner refers to as Yamada's "host"—Yamada's "access module."[10] Yamada's Figure 1 graphically depicts the locations of the error correction unit and CPU within Yamada's controller, not Yamada's "access module" (Petitioner's "host"):

[10] While Yamada does not call its "access module" a "host," Yamada expressly refers to the component of its controller that interfaces with the "access module" as the "host interface unit" or "host I/F unit." *See, e.g.*, Ex. 1005 [Yamada] [0076]-[0077] & Fig. 1. A POSITA would understand Petitioner's reference to "host queries" in Yamada (Pet., 33) to refer to queries made by Yamada's access module. Ex. 2020 [Jones-Decl.] ¶ 77.

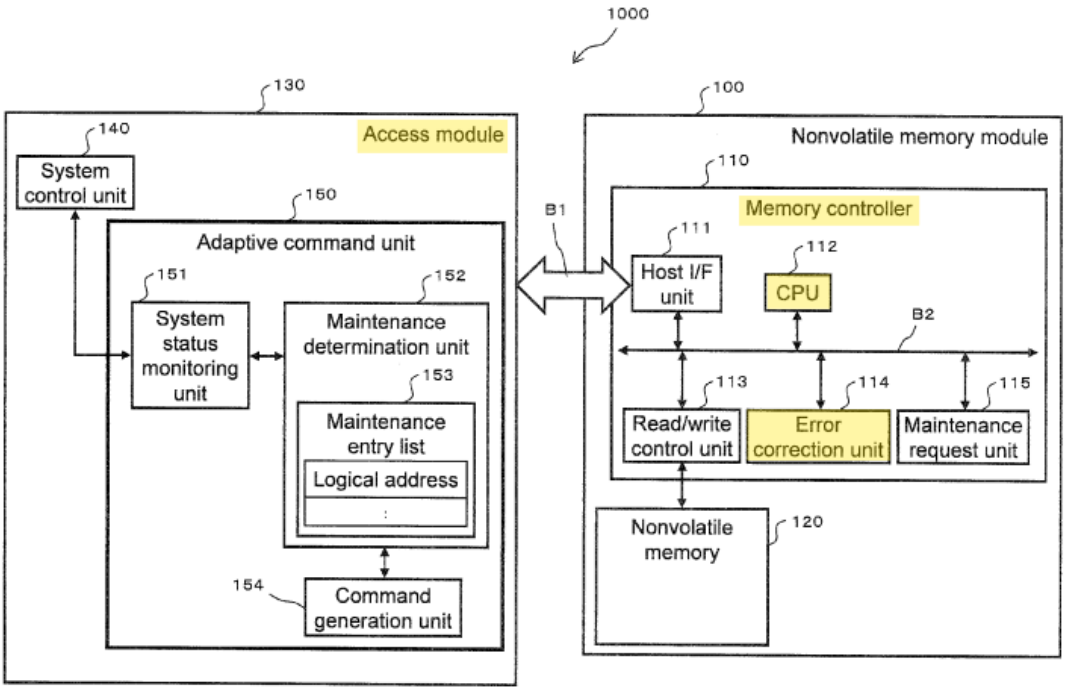


FIG. 1

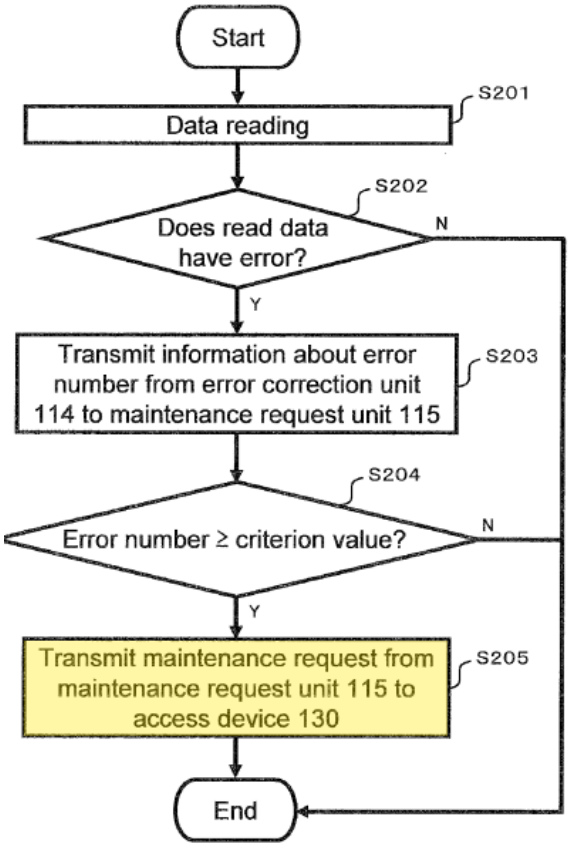
Ex. 1005 [Yamada], Fig. 1 (highlighted by Patent Owner). As Dr. Jones explains:

> When Yamada refers to its "CPU … transmit[ting] the data from the error correction unit 114 to the access module," a POSITA would understand this "transmi[ssion]" to be an activity initiated by the CPU within Yamada's controller. Similarly, in Yamada's paragraph 32, which refers to how Yamada's "maintenance request unit generates a maintenance request" after its error correction unit has corrected an error data, a POSITA would understand that this maintenance request arrives at Yamada's access module by being sent there in activity initiated by the CPU within Yamada's controller.
>
> Indeed, a POSITA would generally understand that this is how maintenance requests in Yamada arrive at Yamada's access module. In the course of generally describing "[t]he internal structure of [its]

> memory controller," Ex. 1005 [Yamada] [0075]-[0082], Yamada states, for example, that "[w]hen determining that the area needs maintenance, the maintenance request unit 115 transmits a maintenance request to the access module 130," *id.*, [0081], and that "[t]he maintenance request is transmitted to the access module 130 via the host I/F unit 111." *Id.*, [0082]. Both the "maintenance request unit 115" and the "host I/F unit 111" are components within Yamada's memory controller; Yamada also states, for example, that "[t]he CPU 112 controls the overall operation of the memory controller 110." *Id.*, [0078].

Ex. 2020 [Jones-Decl.] ¶¶ 78-79. Yamada's "maintenance requests" thus originate at Yamada's ***controller*** and are ***sent*** to Yamada's "access module" without anything resembling a query from the "access module" to produce them. *Id.*

This understanding of the POSITA is confirmed by other disclosures in Yamada, which uniformly refer to the "transmission" of maintenance requests to Yamada's "access module." Yamada's Figure 2, for example, is a flowchart of the process that Yamada's paragraph 96 (cited by Petitioner), and labels the relevant step "Transmit[ting] maintenance request ***from*** maintenance request unit 115 ***to*** access device 130":



Ex. 1005 [Yamada], Fig. 2 & [0096]. All of these steps are "performed by [the] nonvolatile memory module" [memory controller] in Sinclair. *Id.* [0039]; Ex. 2020 [Jones-Decl.] ¶ 81.

Disclosures in Yamada relating to the host-side processing of controller-provided maintenance requests (what Yamada calls its "Data Maintenance Command Generation Cycle") are in accord: this processing begins with the ***receipt*** of a maintenance request ***from*** Yamada's controller, and Yamada never mentions anything like a "query" that initiates this processing. Yamada generally refers to processing as "[t]he procedure performed by the access module 130 ***when receiving***

a maintenance request from the nonvolatile memory module 100," for example, and describes its first step as occurring "[w]hen the access module 130 receives a maintenance request from the nonvolatile memory module 100 (S301)." Ex. 1005 [Yamada] [0100]-[0101]; *see also id.*, Fig. 3 (depicting "Receive maintenance request from nonvolatile memory module 100" as the first step after the "Start" of Yamada's "Data Maintenance Command Generation Cycle"). This feature is consistent across every embodiment where Yamada describes its "access module" processing a maintenance request based on data provided by Yamada's controller. *See, e.g.*, *id.* [0131] (stating that in Figure 6, "the access module 500 ***first receives a maintenance request*** from the nonvolatile memory module 100"), [0156] (similar, for Figure 8).

In sum, there is little beyond Petitioner's *ipse dixit* to support its contention that "Yamada teaches that the controller maintains information about the valid block count and that this information can be retrieved via host queries." *Cf.* Pet. 33. Accordingly, Petitioner does not show any "query received from the host" its combination of Yamada and Sinclair. *In re Magnum Oil Tools*, 829 F.3d at 1380.

III. PETITIONER'S TERASAKI GROUND FAILS (GROUND II).

All of Petitioner's Terasaki-based challenges depend upon Petitioner's Terasaki-Yamada combination as identified in connection with claim 1.[11]

A. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.

1. Petitioner Does Not Say What Its Combination Of Terasaki And Yamada Actually Is.

Petitioner does not identify its combination of Terasaki and Yamada outside of arguments about a particular claim limitation—a limitation relating to a "maintenance request issued by [a] host." Pet. 77-78. At the relevant point in Petitioner's argument, Petitioner has previously identified what it terms an "erase processing request" in Terasaki as the claimed "maintenance request," and is arguing that Terasaki teaches that this "erase processing request" comes from (or at least could obviously come from) Terasaki's host system. *Id.*, 77. In this context, Petitioner implicitly describes its combination of Terasaki with Yamada as follows:

> To the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada. Yamada discloses moving maintenance

[11] The '801 patent has only one independent claim, and Petitioner's arguments for dependent claims all incorporate and rely upon its arguments for claim 1. *See generally* Pet., 85-91 (addressing dependent claims in Ground II).

> operations from the storage controller to its "maintenance determination unit" and "command generation unit" at the host for causing maintenance for data retention, such as refresh, to be performed for an address stored in the maintenance entry list. In the combined Terasaki-Yamada system, the host would issue the erase processing request **[maintenance request issued by the host]** to change zones flagged as "erase requested" to an "erased status."

Id. 77-78 (bracketed insertion and boldface in original) (citations omitted).

None of this identifies what "the combined Terasaki-Yamada system" is. The first sentence of this paragraph is an ***argument*** about a yet-to-be-specified combination. The second sentence of this paragraph is a passing reference to disclosures in Yamada, including specialized components ("maintenance determination unit," "command generation unit") whose operation in ***Yamada*** is not explained.[12] No attempt is made to connect the operation of these components in Yamada to Terasaki. The third sentence is simply a statement that something occurs in "the combined Terasaki-Yamada system."

[12] Petitioner also includes a reproduction of Yamada's Figure 1 with this paragraph. But outside of reproductions of this figure, for example, the phrase "command generation unit" appears only three times in the Petition's discussion of claim 1. Pet., 29 (twice in Sinclair ground), 77 (once in Terasaki ground).

Just as with Petitioner's Sinclair combinations, it is not clear if Petitioner proposes Yamada's "maintenance determination unit" and "command generation unit" would be ***part*** of any alleged "host" in "the combined Terasaki-Yamada system," or if Petitioner is only arguing that Yamada somehow ***teaches*****,** via its disclosures, that a POSITA would modify Terasaki so that Terasaki's as-modified "host" issues what Petitioner terms "erase processing requests." In the former case, as it did with its Sinclair combinations, Petitioner leaves open how Yamada's "maintenance determination unit" receives any information about what alleged "maintenance requests" to issue. Petitioner does not, for example, mention Yamada's controller-side "maintenance request unit," or indeed, mention ***any*** modification to Terasaki's controller as part of its "the combined Terasaki-Yamada system." *See id.*, 78 (Petitioner highlighting only host-side aspects of Yamada's system).

If Petitioner is proposing that Terasaki could be modified in a way that is ***inspired*** by Yamada, but does ***not*** rely on Yamada's specialized components—Petitioner does not identify what those modifications would be. Petitioner's mapping against claim 1 elsewhere identifies certain "write requests from a host," *id.*, 76, and "requests that change the erase flag to an erased flag … [as] initiated by the host," *id.*, 77, as alleged "maintenance requests" in unmodified Terasaki—but does so in connection with arguing that they ***already*** come from Terasaki's host. *See*

id. 76-77. Petitioner does not say how these alleged "maintenance requests" would issue from Terasaki's host in the combined system—except to argue that if they did not already come from Terasaki's host, "the host would issue the … request" in "the combined Terasaki-Yamada system." *Id*., 78. This does not identify "the combined Terasaki-Yamada system," let alone motivate the combination. Petitioner's Terasaki grounds therefore fail for the same reason as its similarly deficient Sinclair grounds do. *See* Section II.A.

2. Petitioner's Alleged Motivation To Modify Terasaki Is Conclusory.

Petitioner's arguments that a POSITA would have been motivated to modify Terasaki are conclusory and should be rejected. *Sisvel Int'l. S.A. v. Sierra Wireless, Inc.*, 82 F.4th 1355, 1364 (Fed. Cir. 2023) (affirming rejection of obviousness rationales that "were too conclusory, lacked clarity, or suffered from both problems"); *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1366–67 (Fed. Cir. 2016) (reversing Board's finding of unpatentability that was "conclusory and unsupported by substantial evidence").

Petitioner's argument that a POSITA would have been motivated to combine Terasaki and Yamada in the way that Petitioner proposes is the following sentence:

> A POSITA would be motivated to combine Terasaki and Yamada for similar reasons as explained in Ground 1 Limitation 1[e], so that a management request, such as erase processing, is issued by the

> host, because as with Sinclair, the prior art includes express teachings, suggestions and motivations to combine the teachings of Terasaki and Yamada such that the erase processing request is issued by the host.

Pet. 78. This is insufficient to state a motivation to combine. *Novartis AG v. Torrent Pharms. Ltd.*, 853 F.3d 1316, 1326 n.2 (Fed. Cir. 2017) (finding that an argument presented by one conclusory sentence "amount[ed] to little more than a request that the Board peruse the cited evidence and piece together a coherent argument on [a party's] behalf"). Petitioner follows its single sentence with a citation to two pages of expert declaration about what Terasaki and Yamada allegedly teach and motivate the POSITA to do, Pet. 78 (citing Ex. 1003 [Litz-Decl.] ¶¶ 212-214), but to any extent that this citation is an attempt to supply new argument incorporated by reference, it would be improper. *See* 37 C.F.R. § 42.6(a)(3).

Petitioner's statement that "[a] POSITA would be motivated to combine Terasaki and Yamada for similar reasons as explained in Ground 1 Limitation 1[e]," moreover, references Petitioner's Sinclair-based Ground I, not a Terasaki ground. Pet. 78. While Petitioner's proffered motivations to combine these references with Yamada arise in connection with the same claim limitation—relating to a "maintenance command issued by [a] host"—the references being combined are not the same. But Petitioner does not engage with their differences. In Sinclair, for example, it is an alleged "flush operation" in Sinclair that provides inspiration for

the alleged "maintenance command issued by a host" in Petitioner's combination, Pet. 30, while in Terasaki, an alleged command "to change zones flagged as 'erase requested' to an 'erased status'" plays that role. Pet. 78. In both cases, Petitioner argues that the primary reference ***already*** discloses that the "host" in the primary reference issues an alleged "maintenance request," *see* Pet. 27 ("Sinclair also discloses that the flush operation may be initiated by the host"), 77 ("Terasaki … discloses that requests that change the erase request flag to an erased flag … may be initiated by the host"), which only raises further questions about how and why a POSITA would be motivated to ***modify*** the references so that alleged "maintenance requests" come from "hosts." But Petitioner does not address these issues—let alone address them in a way that is specific to a combination of Terasaki and Yamada.

Giving identical motivations to combine different references is indicative of hindsight, not a motivation to combine. *See, e.g.*, *UiPath, Inc. v. Rule 14 LLC*, IPR2024-01304, Paper 8, 33 (Mar. 14, 2025) (denying institution) ("Petitioner's reasons to combine Liang and Lucovsky or Mellin are identical to its previously stated reasons to combine Kirsch and Lucovsky or Mellin. While not necessarily improper, repeating the identical argument emphasizes Petitioner's conclusory reasoning that one skilled in the art would simply combine the references for the purposes of adding the feature, entirely independent of the primary references' teachings").

B. Petitioner Does Not Show That Its Terasaki Combination Teaches Any "maintenance request issued by the host."

Claim 1[13] recites a memory controller having logic to, *inter alia*, physically erase memory in fulfilling "a maintenance request issued by the host." Petitioner does not show how this limitation is met in its combination of Terasaki and Yamada.

1. Petitioner Does Not Show Any "maintenance request issued by [a] host" In Terasaki.

The Petition makes several alternative identifications of the claimed "maintenance request issued by a host" in Terasaki. All of them fail.

First, Petitioner states: "during a subsequent write request from the host when maintenance is needed, an erase process is performed **[fulfilling a maintenance request issued by the host]** where the block is changed to an erased block." Pet. 76 (bracketed insertion and boldface in original).

As a threshold matter, it is not clear how Petitioner's juxtaposition of a characterization of Terasaki with claim language is intended to identify an alleged "maintenance request issued by the host." Terasaki never uses the term "write request," for example, but instead repeatedly speaks of write "commands." *See, e.g.*, Ex. 1018 [Terasaki] [0007] (referring to "a write command … received from the host system"), [0025] (referring to "external commands supplied by [the] host system"),

[13] *See supra* note 11.

[0069] (referring to "write processing for data … according to an external command from the host").

Because Petitioner refers to a "write request from the host" in Terasaki, and because claim 1 recites a "maintenance request issued by a host," one might infer that Petitioner is identifying a write command in Terasaki as the claimed "maintenance request from a host." There is an immediate problem with this identification, however. As Dr. Jones explains, a write command in Terasaki does not show the claimed "maintenance request":

> A POSITA would not understand a host-initiated write command in Terasaki's system to be a "maintenance request" in the sense of the '801 patent. Terasaki describes its host-initiated write commands as having two inputs: the data to be written and a range of logical block addresses, *see, e.g.*, Ex. 1018 [Terasaki] [0046] (referring to an exemplary "external command to write data corresponding to a logical block address for LBA15000 to LBA15999 [being] received from the host system"), and a POSITA would understand that such a command in Terasaki is an instruction for Terasaki's controller to simply ***store*** the corresponding data at the corresponding range of logical block addresses. Such a command is not a request for maintenance.

Ex. 2020 [Jones-Decl.] ¶ 94.

The '801 patent reflects a high-level difference between normal input/output and "maintenance," for example, when discussing its Figure 1B embodiment. There,

the ’801 patent describes a “host processor” using “host software … to schedule and manage memory operations, including both ***memory transactions*** and ***memory maintenance*** operations.” Ex. 1001 [’801] 11:16-20; *see also, e.g.*, *id.*, 28:36-38 (distinguishing “commands for memory transactions” from “configuration commands,” “query commands,” and other interactions). Here a POSITA would recognize host-initiated write requests in an embodiment of the ’801 patent like Figure 1B to be examples of “memory transactions” that are not “memory maintenance operations.” Ex. 2020 [Jones-Decl.] ¶ 95.

The ’801 patent also reflects a high-level difference between normal input/output and “maintenance” in its claim language. In one place, for example, claim 1 recites a controller having logic causing the controller to receive “write requests, accompanying addresses and accompanying data from a host,” while in another place, claim 1 recites “a maintenance request issued by the host.” A POSITA would not understand the claimed “write requests” and the claimed “maintenance request” to be literally the same thing, which is what Petitioner is effectively doing when it identifies host write commands in Terasaki with the claimed “write requests … issued by the host,” Pet. 69-70, if Petitioner is separately also identifying host write commands in Terasaki with the claimed “maintenance request issued by a host.” *Id.*, 76; Ex. 2020 [Jones-Decl.] ¶ 96.

Because of Petitioner’s lack of specificity in identifying what it argues the

"maintenance request issued by the host" to be, it is conceivable that Petitioner does not intend host write commands in Terasaki to be synonymous with the claimed "maintenance request issued by a host." It is conceivable, for example, that in saying "a subsequent write request from the host when maintenance is needed, an erase process is performed **[fulfilling a maintenance request issued by the host]**," Pet. 76 (bracketed insertions and boldface in original), Petitioner is not identifying an ***arbitrary*** write command in Terasaki with the claimed "maintenance request issued by a host," but is instead attempting to identify only a write command in Terasaki that occurs "when maintenance is needed" in Terasaki as the claimed "maintenance request issued by a host," or perhaps that only a write command in Terasaki that occurs ***both*** "when maintenance is needed" ***and*** which results in an "erase process [being] performed" as the claimed "maintenance request issued by a host."

As Dr. Jones further explains, these are distinctions without a difference.

> Terasaki does not itself provide a basis for distinguishing a host-issued write command that arrives at Terasaki's controller "when maintenance is needed"—a phrase used by Petitioner that does not appear in Terasaki—from any other write command in Terasaki, such that some host write commands would qualify as "maintenance requests" and others would not. Similarly, Terasaki does not provide a basis for distinguishing host write commands that result in erase operations being performed within Terasaki's controller from host

write commands that do not, such that the former would qualify as "maintenance requests" and the latter would not.

Terasaki never expressly indicates that its host (whose write commands provide only logical block address ranges and data to be written) is even aware that it is interacting with flash storage as distinguished from any other kind of storage. Terasaki also does not describe any way that Terasaki's host could determine in advance which of its write commands would result in erase operations being performed by Terasaki's controller, and which of its write commands would not. The information needed to make such a determination is present in structures like Terasaki's "zone management table" (*see* Ex. 1018 [Terasaki] [0045]-[0047]), but Terasaki locates these structures on its controller and does not describe its host consulting them as a part of issuing write commands.

Ex. 2020 [Jones-Decl.] ¶ 98-99. As Dr. Jones further explains:

Petitioner appears to admit that Terasaki does not disclose its host having access to its "zone management table," for example, because in connection with another limitation, Petitioner relies on another reference, Yamada, to create a system in which the host of the combined system is said to ***literally*** issue "erase processing requests," (that is, not just host write commands that ***result in*** erase processing requests). Pet. 79. Petitioner argues that in this combined system, "the host would maintain Terasaki's zone management table." *Id.* A POSITA would understand that in unmodified Terasaki, the host does not maintain Terasaki's zone management table.

> A POSITA would recognize that in (unmodified) Terasaki, literally the exact same host write command—a command, for example, to write specific data to "a logical block address for LBA10000 through LBA10999" (to use an exemplary of LBAs given in Terasaki, *see id.* [0053]-[0056])—might in some instances result in flash being erased by Terasaki's controller, and in other instances not have this effect. In such circumstances a POSITA would ***not*** understand that some of these write commands were "maintenance requests" in the sense of the '801 patent and others were not. The structure of Terasaki's host write commands does not provide a basis for any such distinction. A POSITA would not recognize ***any*** host write command in Terasaki to be a "maintenance request" in the sense of the '801 patent.

Ex. 2020 [Jones-Decl.] ¶ 100-101.

Perhaps because of these weaknesses in Petitioner's theory of how Terasaki might teach the claimed "maintenance request from a host," Petitioner also offers alternative identifications.

As a second putative "maintenance request from a host," Petitioner states that "Terasaki also discloses that requests that change the erase request flag to an erased flag **[maintenance]** may be initiated by the host **[request issued by the host]**." Pet. 77 (bracketed insertions and boldface in original). The cited support for this assertion is the following high-level description of the "host interface block" in Terasaki's Figure 1:

> Host interface block 7 is a functional block that exchanges data, address information, status information, and external command information with host system 4. In other words, when flash memory system 1 is mounted onto host system 4, flash memory system 1 and host system 4 are connected to each other via external bus 13. In such a state, data and the like that is supplied from host system 4 to flash memory system 1 is taken into the interior of controller 3 using host interface block 7 as an inlet, and data and the like that is supplied from flash memory system 1 to host system 4 is fed into host system 4 using host interface block 7 as an outlet.
>
> In addition, host interface block 7 includes a task file register (not shown in the figure) that temporarily stores host addresses and external commands supplied by host system 4, and an error register (not shown in the figure) that may be set in the event of an error.

Ex. 1018 [Terasaki], ¶¶ 24-25. This disclosure says nothing whatsoever about Terasaki's ***host*** "request[ing] [to] change the erase request flag to an erased flag." *Cf.* Pet. 77 (making this assertion). Accordingly, it does not contain any hint of what "maintenance request issued by a host" Petitioner is arguing to be evidenced by this disclosure.

Petitioner attempts to connect the disclosure to its argument by arguing that because Terasaki discloses "that its controller and host exchange data/addresses and status/command information via host interface block which includes a task file register storing addresses and commands[,] [a] POSITA would understand these

disclosures to teach that the necessary erase processing that is performed on zones flags as 'erase request' is initiated from command information sent by the host to the controller." *Id*. This argument fails for at least two reasons.

First, it is entirely conclusory, supported only by Petitioner's expert (who says the same thing). *Xerox*, IPR2022-00624, Paper 9 at 15.

Second, this argument does not actually engage with the claim language and Petitioner's own mappings of claim 1 onto Terasaki. To summarize, Petitioner has just identified "requests that change the erase request flag to an erased flag" in Terasaki as the claimed "maintenance request[s]." Pet. 77. Petitioner is now attempting to argue that these requests are "issued by [Terasaki's] host," as its mapping would require. *Id*. And instead of showing why the alleged "requests" just identified by Petitioner are, in fact, ***issued by*** Terasaki's host, Petitioner says only that a POSITA would allegedly understand that "***the necessary erase processing*** that is performed on zones flags … is ***initiated from command information sent by the host*** to the controller." *Id*. In speaking in terms of "erase processing" being "initiated" from "command information sent by the host" that has not previously been identified with claim limitations, Petitioner is changing the subject—not even stating, let alone showing, that what Petitioner has identified as the claimed "maintenance request[s]" are "issued by [Terasaki's] host."

At best, Petitioner is hinting that what it has identified as the claimed

"maintenance request[s]" are part of unspecified "processing" that is "initiated from command information sent by the host." This does not show how the limitation is present.

Having failed to show how this limitation is present in Terasaki, Petitioner turns to Yamada. Pet. 77. This argument also fails.

2. Petitioner Does Not Show That Yamada Teaches Any "maintenance request issued by the host" In Its Combination.

Petitioner argues "[t]o the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada." Pet. 77. As a threshold matter, it is not initially clear what Petitioner means by "the erase processing request": at this stage in its argument, Petitioner has previously identified both "requests that change the erase request flag to an erased flag" (Pet. 77) and "a subsequent write request from the host when maintenance is needed," perhaps as a result of which "an erase process is performed" (Pet. 76) as alleged "maintenance request[s] issued by the host." It is not clear which of these two earlier examples is Petitioner's "the erase processing request."

Petitioner subsequently states that "[i]n the combined Terasaki-Yamada system, the host would issue the erase processing request [**maintenance request issued by the host**] to change zones flagged as 'erase requested' to an 'erased

status.'" Pet. 78 (bracketed insertion and boldface in original).[14] With this disclosure, Petitioner presumably identifies a command "to change zones flagged as 'erased requested' to an 'erase status'" in Terasaki—or perhaps a command to erase a Terasaki zone that has the ***result*** of changing a zone flagged as "erase requested" to a zone flagged "erased"—as the claimed "maintenance request issued by the host" in its argument.

But Petitioner does not link this command in ***Terasaki*** to any structure in "the combined Terasaki-Yamada system." Indeed, Petitioner does not identify what "the combined Terasaki-Yamada system" is. *See, e.g.*, Section III.A.1 *supra.* Petitioner's entire argument about the combination is a single paragraph.

> To the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada. Yamada discloses moving maintenance operations from the storage controller to its "maintenance determination unit" and "command generation unit" at the host for causing maintenance for data retention, such as refresh, to be performed for an address stored in the maintenance entry list. In the combined Terasaki-Yamada system, the host would issue the erase processing

[14] Terasaki does not use the phrase "erased status," but uses the word "erased" to describe a "status" of a zone. *See, e.g.*, Ex. 1018 [Terasaki], Fig. 8, [0045].

> request **[maintenance request issued by the host]** to change zones flagged as "erase requested" to an "erased status."

Pet. 77-78 (bracketed insertion and boldface in original) (citations omitted).

Stripped of its argumentative assertions (the first and last sentences), this paragraph is a partial description of two components disclosed in Yamada—a "maintenance determination unit" and a "command generation unit." Petitioner does not say whether "the combined Terasaki-Yamada system" contains ***copies*** of these components, or whether "the combined Terasaki-Yamada system" is a modified version of Terasaki that ***adopts the functionality*** of these components (let alone how that would be done). Indeed, Petitioner does not describe what the combined system would be, other than to say it would contain the claimed "maintenance request issued by the host." This is conclusory, and does not show the limitation. *In re Magnum Oil Tools*, 829 F.3d at 1380.

C. Petitioner Does Not Show That Its Terasaki Combination Teaches Logic To Cause A Memory Controller To "identify a specific one of the respective subdivisions . . . ."

Claim 1[15] recites, *inter alia*, a memory controller having logic to cause the memory controller to "receive write requests, accompanying addresses and accompanying data from a host and, with respect to each one of the write requests,

[15] *See supra* note 11.

[to] identify a specific one of the respective subdivisions dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests." Petitioner identifies a "write command" in Terasaki, which may include a logical address and data, with the claimed "write request[]" and "accompanying address[] and accompanying data," Pet. 69-70, and the claimed "subdivisions" with Terasaki's "zones." *Id.*, 67-68. But Petitioner does not show how Terasaki's controller allegedly "identif[ies] a specific one of the respective subdivisions" [a Terasaki zone, in Petitioner's read] "dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests." Petitioner does not refer to Yamada in argument about this limitation. *See id.*, 70-71.

Petitioner states that "Terasaki discloses that a corresponding LBA indicates which LBA the block corresponds to when data is stored in the block." *Id.*, 70. But Petitioner does not link this disclosure to any identification of "a specific one" of any Terasaki "zones" from an address accompanying a write command. Petitioner then says that "Terasaki discloses a zone management table that manages the storage status of each zone and manages the LBA of the data written onto each zone by zone, as show[n] in Fig. 6." Pet. 70-71. Again, however, Petitioner does not link this table to the identification of "a specific one" of any Terasaki "zone" from any address accompanying a Terasaki write command, let alone show Terasaki doing this

"dependent on which one of the non-overlapping ranges encompasses" an address accompanying a write command. Petitioner states that "[t]his initial stored data is written onto the zone management table above for the following LBAs," and lists an exemplary correspondence between Terasaki "zones" and ranges of LBAs in Terasaki's Figure 6. *Id.* But again, this does not show the limitation. As Dr. Jones explains:

> Terasaki's Figure 6 depicts a correspondence between exemplary Terasaki "zones" and certain ranges of LBAs in Terasaki, but does not support Petitioner's contention that Terasaki's controller somehow uses a host-supplied ***address*** from a write command in Terasaki to identify a Terasaki "zone" to write to. As Petitioner appears to acknowledge, Terasaki describes its controller processing a host command to write to a logical address by using its "zone management table" to find a Terasaki zone in an "erased" status, and writing the data to that Terasaki zone. *See, e.g.*, Ex. 1018 [Terasaki] [0046], Pet. 72 (citing paragraph 46 of Terasaki's discussion of its Figure 6).
>
> Terasaki does not disclose any selection rule by which its controller selects a ***particular*** "erased" Terasaki zone to write to. As Petitioner elsewhere describes this process, "an empty erased zone is found within the zone management table," Pet. 72; Terasaki similarly only states that upon a host write request, an "erased zone will be found within the zone management table, and the data will be written onto that erased zone." Ex. 1018 [Terasaki] [0046]. A POSA would not understand from these disclosures that an address accompanying the

> host write request is necessarily or even advantageously used in selecting an erased zone to write to. It would be consistent with Terasaki, for example, for the controller to select a zone on a basis entirely unrelated to any address supplied by the host—such as simply selecting the lowest-numbered zone that is in an "erased" status, or selecting an erased zone on the basis of wear information (such as Terasaki's "erase count . . . for each zone," which Terasaki expressly suggests could be used "to manage the order of priority of the erased zones to be used," *id.* [0049]). Terasaki gives an example in which when LBA 11000 is rewritten, it is written to Zone 7 when it was previously in Zone 2—another indication that the zone chosen for writing does not depend on the LBA. *Id.* [0048].

Ex. 2020 [Jones-Decl.] ¶ 115-116. For at least these reasons, Petitioner fails to show how this limitation is present in its Terasaki combination.

D. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller "stor[ing] a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host."

Claim 1[16] recites, *inter alia*, a memory controller having logic to cause the memory controller to "store a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host."

[16] *See supra* note 11.

Petitioner's discussion of "a value representing a size of the NAND flash memory that is to be physically erased" does not identify where this limitation is said to appear in Terasaki. Yet a petition must state its grounds "with particularity," 35 U.S.C. § 312(a)(3), and "must specify where each element of the claim is found in the prior art patents or printed publications relied upon." 37 C.F.R. § 42.104(b)(4). Petitioner's Terasaki-based challenges fail to meet this standard.

In connection with identifying the claimed "value," Petitioner states:

> When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet. 76 (bracketed insertion and boldface in original) (citing Ex. 1003 [Litz-Decl.] ¶ 239) (citations omitted). Merely placing the claim limitation in brackets next to a statement about what happens "[w]hen data is rewritten" does not clearly identify what Petitioner contends is a "value representing a size of the of the NAND flash memory that is to be physically erased"—is it the "1000" of 1000 blocks? Is it the "up to 1000 blocks" (and which blocks, exactly)? Is it the status of the "erase request flag" ***for*** the up to 1000 blocks (playing the role of a potentially non-numerical "value")? In what sense do any of these quantities represent a "size of the NAND flash memory that is to be physically erased," and how or where does Terasaki's

controller "store" them?

The paragraph of expert declaration that Petitioner cites in connection with this limitation does not resolve this issue. It states, in relevant part:

> When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data was written is changed to an erase request flag for the up to 1000 blocks of that one zone, ***which represents a size of the NAND flash memory that is to be physically erased***. EX1018, ¶68. This erase request flag indicates that erase processing for that zone is necessary. *Id.*

Ex. 1003 [Litz-Decl.] ¶ 239.

The paragraph of Terasaki cited by both Petitioner and Petitioner's expert does not shed light on this issue either. It states:

> If the data is rewritten, the zone erase process described above will be performed on the zone in which the original data was written. Therefore, if the data is rewritten, it is preferable to change the written flag written on the first page of the initial good block in the zone in which the original data was written to the erase request flag, and then to change the erase request flag to the erased flag when the above-noted zone erase processing has been performed.

Ex. 1018 [Terasaki] [0068].

The knowledge of the POSITA does not resolve the ambiguities presented by Petitioner. As Dr. Jones explains:

I am not certain of what Petitioner proposes to identify as the "value representing a size of the NAND flash memory that is to be physically erased." As one example, I do not understand Petitioner's use of the qualifier "for the up to 1000 blocks of that one zone" in the statement that "the written flag … in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone." Pet. 76. A POSITA would understand from Terasaki's various express disclosures that the "erase request" flag in Terasaki is set on a zone basis, not on the basis of individual blocks within a zone. *See, e.g.*, Ex. 1018 [Terasaki] [0047] ("when revising data that has already been written, the zone management table … will change the storage status ***of that zone*** to an erase request flag (a flag indicating that erase processing is necessary to convert ***the zone*** to ***an erased zone***)"), [0049] (referring to "information indicating the state of each zone (write state, erase state, erase request state, etc.)"). This is depicted graphically, for example, in Terasaki's Figure 8, which depicts a "Zone 2 in which the original data was written … changed to an erase request flag." *Id.* [0048].

(FIGURE 8)

Zone	Logical block address
Zone 1	LBA10000
Zone 2	Erase request
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	LBA15000
Zone 7	LBA11000
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

Id., Fig. 8 (highlighting added). Thus, when Petitioner speaks of changing a flag "to an erase request flag for the up to 1000 blocks of that one zone," for purposes of identifying the claimed "value representing a size of the NAND flash memory that is to be physically erased," Petitioner's meaning is unclear, as the erase request flag, if it is set, is set for the entire zone—not for individual blocks within the zone, and not for "up to 1000 blocks of that one zone."

Ex. 2020 [Jones-Decl.] ¶ 123.

"It is not the Board's job to cobble together assertions from different sections of a petition … to infer every possible permutation of a petitioner's arguments. Arguments in a petition must be made with particularity, not opacity." *See Microsoft Corp. v. FG SRC, LLC*, 860 F. App'x. 708, 713 (Fed. Cir. 2021) (nonprecedential);

see also Netflix, Inc. v. DivX, LLC, 84 F.4th 1371, 1377 (Fed. Cir. 2023) ("A petitioner may not rely on a vague, generic, and/or meandering petition and later fault the Board for failing to understand what the petition really meant. Ultimately, it is the petitioner's burden to present a clear argument."). By not even identifying a purported "value" in the cited art, Petitioner's Terasaki-based challenges fail.

E. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller "provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value."

Claim 1[17] recites, *inter alia*, a "storage device" comprising a "memory controller" having logic to cause the memory controller to "provide to the host, responsive to a ***query received from the host***, the information, as updated, and the stored value." Petitioner does not show how this limitation is taught in its combination of Terasaki and Yamada.

Petitioner identifies Terasaki's memory controller with the claimed "memory controller," Pet. 67, and Terasaki's host with the claimed "host." *Id.*, 69. For the claimed "information," Petitioner identifies "information indicating the state of each zone (write state, erase state, erase request state, etc.)" and "information on the data stored in each zone (information on the physical block address of the block where

[17] *See supra* note 11.

the data is stored, information on the logical block address of the stored data, etc.)." *Id.*, 75 (quoting Ex. 1018 [Terasaki] [0049]). As noted above, *see* Section III.D *supra*, Petitioner does not expressly identify any "stored value" in Terasaki, but states:

> When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet. 76 (bracketed insertion and boldface in original).

With these identifications, Petitioner does not explain how Terasaki's controller "provide[s] to the host, responsive to a query received from the host, the information, as updated, and the stored value," for at least three reasons: Petitioner never identifies anything in Terasaki as the claimed "query received from [a] host," Petitioner does not link "the information" recited in this limitation to what it has previously identified as "the information" in other limitations, and Petitioner does not identify what claimed "stored value" is allegedly "provid[ed] to the host."

In more detail, Petitioner argues as follows:

> Terasaki discloses that the controller and host exchange data, address information, status information [**provide to the host… the stored value**], and command information via host interface block 7

> which includes a task file register storing addresses and commands, and an error register that may be set in the event of an error. Specifically, status information is "fed into host system 4 using host interface block 7 as an outlet." For instance, Terasaki discloses that to pass error information to the host, the host interface block that is used to pass information from the controller to the host may include an error register that may be set in the event of an error.

Pet. 79 (bracketed insertions and boldface in original) (citations omitted).

First, this argument fails to identify any alleged "query received from a host." The relevant limitation involves a controller "provid[ing] to the host, responsive to a ***query received from the host***, the information, as updated, and the stored value." Petitioner's statement that "the controller and host ***exchange*** data, address information, status information, and command information" does not identify any alleged "query received from the host" in Terasaki. Indeed, the claim term "query" does not appear anywhere in Petitioner's argument (or indeed, anywhere in Petitioner's Terasaki ground). If this is an attempt at an inherency argument, *see, e.g.*, Section II.B.1 *supra* (discussing inherency), it fails because a POSITA would recognize that Terasaki's controller and Terasaki's host can exchange information without a "query received from [its] host." Ex. 2020 [Jones-Decl.] ¶ 129.

Second, "the information" in the claimed "***provid[ing] to the host***, responsive to a query received from the host, ***the information***" is a reference back to earlier-

recited "information, for the specific one of the respective subdivisions, representing an extent to which the erase blocks of the specific one of the respective subdivisions are full." Ex. 1001 ['801] cl. 1. Petitioner previously identified "information indicating the state of each zone (write state, erase state, erase request state, etc.)" and "information on the data stored in each zone (information on the physical block address of the block where the data is stored, information on the logical block address of the stored data, etc.)" in Terasaki as the claimed "information." Pet. 75. But here, Petitioner does not even argue, let alone show, that this earlier-identified "information" is provided to Terasaki's host. If Petitioner's vague statement that "status information is 'fed into host system 4 using host interface block 7 as an outlet,'" *id.*, 79 (quoting Ex. 1018 [Terasaki] [0024]), is an implicit reference back to Petitioner's earlier-identified "information," the cited support for Petitioner's argument—Terasaki's statement that "status information" is "fed into host system 4"—does not prove the point. This statement comes from a general disclosure in Terasaki that reads as follows:

> Host interface block 7 is a functional block that exchanges data, address information, status information, and external command information with host system 4. In other words, when flash memory system 1 is mounted onto host system 4, flash memory system 1 and host system 4 are connected to each other via external bus 13. In such a state, data and the like that is supplied from host system 4 to flash

> memory system 1 is taken into the interior of controller 3 using host interface block 7 as an inlet, and data and the like that is supplied from flash memory system 1 to host system 4 is fed into host system 4 using host interface block 7 as an outlet.

Ex. 1018 [Terasaki] [0024]. If this paragraph's references to "data and the like" support some argument that Petitioner's earlier-identified "information" is "provide[d] to the host, responsive to a query received from the host," it is not an argument that Petitioner makes. But "the Board should [] not have to decode a petition to locate additional arguments beyond the ones clearly made." *Netflix*, 84 F.4th at 1377.

Third and finally, "the stored value" in the claimed "***provid[ing] to the host***, responsive to a query received from the host … ***the stored value***" is a reference back to earlier-recited "value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host." As noted above, *see* Section III.C *supra*, Petitioner never identifies a "value" in Terasaki, but only states:

> When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet. 76 (bracketed insertion and boldface in original). Even if one assumes, *arguendo*, that this ***did*** identify a "value" in Terasaki, Petitioner's argument in connection with the present limitation makes no reference to any "erase request flags" or blocks in zones in which data was rewritten. *See id.*, 79. Petitioner is asking the Board to sift through Terasaki and find support for an argument that it has not itself made. The Board should reject this invitation. *Sisvel Int'l. S.A. v. Sierra Wireless, Inc*., 82 F.4th 1355, 1364 (Fed. Cir. 2023) (affirming rejection of obviousness rationales that "were too conclusory, lacked clarity, or suffered from both problems").

Finally, Petitioner argues that this limitation is obvious in view of Yamada if it is not already taught by Terasaki, in two conclusory sentences:

> As explained in 1[e], in the combined Terasaki-Yamada system, the host would issue the erase processing request. A POSITA would understand these disclosures to teach that in such an embodiment the host would maintain Terasaki's zone management table including the updated status of each zone, in order for the host to successfully perform this task.

Pet. 79. This argument fails because it is conclusory, because it is only supported by an expert declaration that repeats it, *see* Ex. 1003 [Litz-Decl.] ¶ 247, and because—like Petitioner's argument in view of Terasaki alone—it fails to identify any "query

from the host," or refer back to what Petitioner earlier identified as the claimed "value."

IV. CONCLUSION

For the foregoing reasons, the Board should not institute trial.

Respectfully submitted,

Dated: November 5, 2025

By: /s/ *Kenneth Weatherwax*
Kenneth Weatherwax (Reg. No. 54,528)
Nathan Lowenstein (*pro hac vice*)
Dennis Courtney (*pro hac vice*)
Colette Woo (*pro hac vice*)
LOWENSTEIN & WEATHERWAX LLP

Xu Zhou (*pro hac vice*)
CALDWELL CASSADY CURRY PC

Attorneys for Patent Owner

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITS

This Patent Owner Preliminary Response (the "POPR") consists of 13954 words, excluding table of contents, table of authorities, certificate of service, this certificate, or table of exhibits. The POPR complies with the type-volume limitation of 14,000 words as mandated in 37 C.F.R. § 42.24. In preparing this certificate, counsel has relied on the word count of the word-processing system used to prepare the paper (Microsoft Word).

Respectfully submitted,

Dated: November 5, 2025

By: /s/ *Dennis Courtney*
Dennis Courtney
Attorney for Patent Owner

CERTIFICATE OF SERVICE

The undersigned hereby certifies that the following documents were served by electronic service, by agreement between the parties, on the date below:

PATENT OWNER'S PRELIMINARY RESPONSE

EXHIBIT 2019

EXHIBIT 2020

The names and addresses of the parties being served are as follows:

James M. Glass	jimglass@quinnemanuel.com
John T. Mckee	johnmckee@quinnemanuel.com
Luiz Miranda	luizmiranda@quinnemanuel.com

Respectfully submitted,

Dated: November 5, 2025 By: /s/ *Michael Cosley*
Michael Cosley