

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 IPR2025-01266
Patent 11,544,183

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	PETITIONER DOES NOT SHOW A MOTIVATION TO COMBINE WITH A REASONABLE EXPECTATION OF SUCCESS.	2
A.	Petitioner’s Ellis Combinations Are Underspecified (Grounds I–II, All Challenged Claims).....	2
B.	Petitioner’s Sinclair Combinations Are Underspecified (Grounds III- IV, All Challenged Claims).	10
III.	PETITIONER’S ELLIS GROUNDS FAIL (GROUNDS I–II, ALL CHALLENGED CLAIMS).....	13
A.	Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.”	14
1.	Ellis Does Not Teach Or Render Obvious A Write Request “from a host . . . accompanied by an address that designates a specific one of the subdivisions.”	14
2.	Son Does Not Teach Or Render Obvious A Write Request “from a host . . . accompanied by an address that designates a specific one of the subdivisions” In Petitioner’s Combination.....	18
B.	Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.”	22
1.	Ellis Does Not Teach Sending “information indicating extent of page utilization of [a] specific one of [its] subdivisions” To The Alleged “host” In Petitioner’s Combinations.	24

2.	Son Does Not Teach Or Render Obvious Sending “information indicating extent of page utilization of [a] specific one of [its] subdivisions” To The Alleged “host” In Petitioner’s Combinations.....	28
C.	Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “an erasure request from the host . . . the erasure request accompanied by the address corresponding to the specific one of the subdivisions” (Grounds I-II, All Challenged Claims).	32
1.	Petitioner Does Not Show Erasure Requests From Ellis’s Host.	33
2.	Petitioner Does Not Show That A Host-Initiated Erase Request Assumed To Exist In Ellis Would Be “accompanied by [an] address corresponding to the specific one of the subdivisions.”	36
3.	Son Does Not Teach Or Render Obvious The Claimed Erasure Request In Petitioner’s Combination.....	38
IV.	PETITIONER’S SINCLAIR GROUNDS FAIL (GROUND III–IV, ALL CHALLENGED CLAIMS)	40
A.	Petitioner Does Not Show That Its Sinclair Combinations Render Obvious “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions” (Grounds III-IV, All Challenged Claims).	40
1.	Sinclair Does Not Teach The Claimed “write request from a host . . . accompanied by an address that designates a specific one of the subdivisions.”	41
2.	Sinclair Does Not Render Obvious The Claimed ““write request from a host ... the write request accompanied by an address that designates a specific one of the subdivisions” In Petitioner’s Combinations.....	48

B. Petitioner Does Not Show That Its Sinclair Combinations Render Obvious “logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.” 49

C. Petitioner Does Not Show That Its Sinclair Combinations Teach Or Render Obvious “an erasure request from the host . . . the erasure request accompanied by the address corresponding to the specific one of the subdivisions.” 55

V. CONCLUSION 58

TABLE OF AUTHORITIES

Page(s)

COURT DECISIONS

<i>ActiveVideo Networks, Inc. v. Verizon Commc’ns, Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	53
<i>Belden Inc. v. Berk-Tek LLC</i> , 805 F.3d 1064 (Fed. Cir. 2015)	6
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	4
<i>Icon Health & Fitness, Inc. v. Strava, Inc.</i> , 849 F.3d 1034 (Fed. Cir. 2017)	42
<i>In re Kotzab</i> , 217 F.3d 1365 (Fed. Cir. 2000)	3, 10
<i>In re Kubin</i> , 561 F.3d 1351 (Fed. Cir. 2009)	7
<i>In re Magnum Oil Tools Int’l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	43, 55
<i>KSR Int’l Co. v. Teleflex Inc.</i> , 550 U.S. 398 (2007).....	8
<i>PAR Pharm. v. TWi Pharms., Inc.</i> , 773 F.3d 1186 (Fed. Cir. 2014)	36
<i>PersonalWeb Techs. LLC v. Apple, Inc.</i> , 848 F.3d 987 (Fed. Cir. 2017)	2
<i>Southwire Co. v. Cerro Wire LLC</i> , 870 F.3d 1306 (Fed. Cir. 2017)	36

<i>Trintec Indus., Inc. v. Top U.S.A. Corp.</i> , 295 F.3d 1292 (Fed. Cir. 2002)	36
---	----

<i>Virtek Vision Int’l ULC v. Assembly Guidance Sys.</i> , 97 F.4th 882 (Fed. Cir. 2024)	6
---	---

AGENCY DECISIONS

<i>Apple, Inc. v. Voip-pal.com, Inc.</i> , IPR2016-01198, Paper 53 (Nov. 20, 2017)	7
---	---

<i>DISH Network LLC v. Entropic Commc’ns, Inc.</i> , IPR2024-00462, Paper 14 (Aug. 26, 2024)	4
---	---

<i>Johns Manville Corp. v. Knauf Insulation, Inc.</i> , IPR2018-00827, Paper 9 (Oct. 16, 2018) (informative)	22
---	----

<i>Microsoft Corp. v. Virtru Corp.</i> , IPR2023-00019, Paper 19 (May 17, 2023)	3
--	---

<i>Peddinghaus Corp. v. Ficep Corp.</i> , IPR2020-01707, Paper 15 (May 11, 2021)	34
---	----

<i>Samsung Elecs. Co., LTD v. Red Rock Analytics, LLC</i> , IPR2018-00556, Paper 18 (Aug. 20, 2018)	54
--	----

<i>Samsung Elecs. Co., Ltd. v. Dynamics Inc.</i> , IPR2020-00499, Paper 63 (Aug. 10, 2021)	9
---	---

<i>UiPath, Inc. v. Rule 14 LLC</i> , IPR2024-01304, Paper 8 (Mar. 14, 2025)	12
--	----

<i>Vectair Sys., Inc. v. Fresh Prods., Inc.</i> , IPR2024-00824, Paper 9 (Nov. 12, 2024)	43
---	----

<i>Vicor Corp. v. Delta Elecs., Inc.</i> , IPR2024-00715, Paper 13 (Oct. 2, 2024)	34
--	----

<i>Xerox Corp. v. Bytemark, Inc.</i> , IPR2022-00624, Paper 9 (Aug. 24, 2022) (precedential)	44, 49
---	--------

REGULATIONS

37 C.F.R. § 42.6(a)(3).....44

37 C.F.R. § 42.65(a)..... 44, 49

EXHIBIT LIST	
2001	<i>Reserved</i>
2002	<i>Reserved</i>
2003	<i>Reserved</i>
2004	<i>Reserved</i>
2005	Declaration of Mike Jadon, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2-24-cv-01073 (E.D. Tex. May 6, 2025), Dkt. 43-13 [Jadon-Decl.]
2006	Laura Caulfield, <i>Project Denali to define flexible SSDs for cloud-scale applications</i> , 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, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 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 <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 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, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2-24-cv-01073 (E.D. Tex. July 11, 2025), Dkt. 64 [Order]
2011	Plaintiff's Opposed Motion to Dismiss, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2-24-cv-01073 (E.D. Tex. July 17, 2025), Dkt. 69 [Motion-to-Dismiss]

2012	First Amended Docket Control Order, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 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, <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 2-24-cv-01073 (E.D. Tex. June 18, 2025 [Invalidity-Contentions]
2014	Statement of Interest of the United States of America <i>Radian Memory Systems LLC v. Samsung Electronics Co., Ltd. et al.</i> , 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	<i>Reserved</i>
2018	Median and Average Time To Trial for Judge Rodney Gilstrap, E.D. Tex., Docket Navigator [Gilstrap Time to Trial]
2019	<i>Reserved</i>
2020	Declaration of Dr. Mark Jones, Ph.D. [Jones-Decl.]
2021	U.S. Patent No. 8,949,689
2022	U.S. Patent No. 9,021,231

2023	U.S. Patent No. 9,098,399
2024	U.S. Patent No. 9,123,445
2025	U.S. Patent No. 9,170,941
2026	U.S. Patent No. 9,183,137
2027	U.S. Patent No. 9,244,519
2028	U.S. Patent No. 9,543,025
2029	U.S. Patent No. 9,671,962
2030	U.S. Patent No. 10,546,648

All emphases in quotations are added unless otherwise stated.

I. INTRODUCTION

The Board should deny institution because Petitioner fails to show a reasonable likelihood of prevailing on any challenged claim.

Petitioner’s Ellis-based grounds rest on a hindsight-driven attempt to merge the architectures of Ellis and Son, despite their conflicting allocations of flash-management responsibility: Ellis assigns the work of logical-to-physical translation, wear leveling, garbage collection, and related operations to its controller, while Son assigns similar tasks to its “SSD module” (which Petitioner characterizes as a host). Petitioner’s cursory treatment of the allocation of responsibility between host and controller in its combination fails to reconcile the differences between these architectures. Petitioner’s Sinclair-based grounds are similarly unmotivated and hindsight-driven, and attempt to incorporate aspects of Petitioner’s (deficient) Ellis-based grounds simply by citing those grounds.

Beyond failing to provide a motivation to combine, Petitioner fails to show that the cited art teaches key limitations. These omissions are not cured by Petitioner’s reliance on boilerplate text, conclusory statements, and cross-incorporated arguments—features of the Petition that only highlight the fact that Petitioner has not carried its burden. For reasons articulated more fully below, the Board should therefore deny institution.

II. PETITIONER DOES NOT SHOW A MOTIVATION TO COMBINE WITH A REASONABLE EXPECTATION OF SUCCESS.

Petitioner’s 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[] explan[ation]” 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.*

A. Petitioner’s Ellis Combinations Are Underspecified (Grounds I–II, All Challenged Claims).

Rather than describing exactly how Ellis would be modified to incorporate alleged teachings of Son, Petitioner provides only a high-level summary of aspects of the *outcome* of a modification, stating for example that “a POSA would have been motivated to incorporate Son’s teaching regarding (i) additional forms of superblock metadata and (ii) the sharing of superblock metadata with a host for purposes of memory management” into Ellis, Pet., 9, or “to combine Son’s host functionality with the system described in Ellis.” Pet., 10.

Although Petitioner’s arguments repeatedly cite Son’s host-side “SSD

module” for disclosures relating to the alleged “host” of its combination, Petitioner never explains how Son’s “SSD module” is actually used to modify Ellis’s host, or how Ellis’s memory controller is modified to interact with that host. Petitioner effectively asks the Board to presume that core aspects of its combination operate to satisfy the challenged claims. But merely pointing to disparate features in the art is insufficient to state a combination. *Microsoft Corp. v. Virtru Corp.*, IPR2023-00019, Paper 12, 18 (May 17, 2023) (denying institution) (“Nor does Petitioner explain why it would have been obvious to use both mechanisms in the process of validating one user, and obviousness cannot be shown merely by pointing to disparate features of a disclosure”); *see also In re Kotzab*, 217 F.3d 1365, 1371 (Fed. Cir. 2000) (“a rejection cannot be predicated on the mere identification in [a prior art reference] of individual components of claimed limitations”).

As one example, Petitioner relies upon “additional forms of superblock metadata” from Son to buttress an argument that its combination of Ellis and Son teaches the claimed memory controller with logic to receive the claimed “write request.” *See* Pet., 14-15. In doing so, Petitioner blends characteristics of distinct components (such as Ellis’s “host system,” Son’s “SSD module,” and the controllers that these systems employ) without fully addressing which responsibilities are undertaken by which elements of its combination.

Petitioner never quite argues, for example, that its combination would contain

both a copy of Ellis’s “host system” and a copy of Son’s “SSD module” within what it alleges is the claimed “host” of claim 1, or that Ellis’s controller would **stop** performing duties in the combination that Petitioner suggests the alleged teachings of Son’s “SSD module” would allow the “host” to perform. Simply stating that one reference adds “additional types of metadata” to the other does not, by itself, address **which** functions of Ellis are retained in the combination and which are modified in view of Son. Many aspects of the combination are left to inference from element-by-element mapping—an approach indicative of impermissible hindsight. *See, e.g., DISH Network LLC v. Entropic Commc’ns, Inc.*, IPR2024-00462, Paper 14, 28 (Aug. 26, 2024) (denying institution) (noting that “care must be taken to avoid hindsight reconstruction by using the patent in suit as a guide through the maze of prior art references, combining the right references in the right way so as to achieve the result of the claims in suit”) (cleaned up). Relying on an underspecified combination is sufficient reason to reject a ground. *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (“it was [petitioner’s] burden to explain to the Board how Haskell combined with Rossmere rendered the challenged claims unpatentable”).

Another example of Petitioner’s difficulties with its combination is given by Petitioner’s argument that its Ellis combinations render obvious the claimed memory controller with “logic to send the host the information indicating extent of page utilization.” Pet., 16-17. Ellis does not disclose its memory controller sending its

host system what Petitioner identifies as “information indicating extent of page utilization” in Ellis. *See* Section III.B *infra*. Petitioner encounters a similar problem with the claimed “logic to receive an erasure request from the host” (Pet., 18), as Ellis does not disclose its host system sending anything that Petitioner can identify with the claimed “erasure request.” *See* Section III.C *infra*. In both situations, Petitioner attempts to bridge the evident gap between Ellis and the claims by citing the following:

The functional units in the memory controller 104 can work individually and independently of the other functional units. For illustrative purposes, the memory controller 104 is described by operation of the memory controller 104 with the host system 108 and the memory array 106. ***It is understood that the memory controller 104, the host system 108, and the memory array 106 can operate any of the modules and functions of the memory controller 104.***

Ex. 1004 [Ellis] 4:58-65; *see* Pet. 17, 18 (citing this disclosure in connection with claim limitations); *see also* Pet. 10, 33 (citing this disclosure in connection with motivation to combine).

In the context of Petitioner’s alleged motivation to combine Ellis and Son, Petitioner frames this disclosure as a teaching that a POSITA would have been motivated to move essentially any functionality in Ellis from Ellis’s controller to Ellis’s host. (The language Petitioner specifically uses is that Ellis’s column 4

teaches that “memory management responsibility may be shared between the controller and the host.” *See* Pet., 10, 33.)

As a threshold matter, a high-level disclosure of an array of alternatives—*e.g.*, assigning controller ‘modules and functions’ to the controller, host, or memory—does not show why a POSITA would choose any particular set of alternatives, much less do so with a reasonable expectation of success. *See, e.g., Virtek Vision Int’l ULC v. Assembly Guidance Sys.*, 97 F.4th 882, 886-887 (Fed. Cir. 2024) (“It does not suffice to meet the motivation to combine requirement to recognize that . . . alternative arrangements . . . were . . . known in the art”); *see also, e.g., Belden Inc. v. Berk-Tek LLC*, 805 F.3d 1064, 1073 (Fed. Cir. 2015) (“obviousness concerns whether a skilled artisan not only *could have made* but *would have been motivated to make*”) (emphasis in original).

The probative value of this paragraph in Ellis should also be assessed in context: the same disclosure appears, verbatim or near verbatim, in at least ten unrelated patent applications filed by the initial assignee of Ellis, most of which list other sets of named inventors.¹ Such text—a boilerplate reminder that the scope of

¹ *See, e.g.*, Ex. 2021 [U.S. Patent No. 8,949,689] 4:32-39; Ex. 2022 [U.S. Patent No. 9,021,231] 4:60-67; Ex. 2023 [U.S. Patent No. 9,098,399] 4:66–5:6; Ex. 2024 [U.S. Patent No. 9,123,445] 4:43-49; Ex. 2025 [U.S. Patent No. 9,170,941]

a patent disclosure is not generally limited to the specifics of disclosed embodiments—does not analyze any trade-offs, performance implications, or engineering considerations that might inform whether a host-based or controller-situated functionality is preferable in any system. In particular, such text does not motivate Petitioner’s specific modifications to the system that Ellis actually describes. *See, e.g., In re Kubin*, 561 F.3d 1351, 1359 (Fed. Cir. 2009) (“where a defendant merely throws metaphorical darts at a board filled with combinatorial prior art possibilities, courts should not succumb to hindsight claims of obviousness.”); *Apple, Inc. v. Voip–pal.com, Inc.*, IPR2016-01198, Paper 53 (Final Written Decision) at 23 (Nov. 20, 2017) (“Petitioner’s position ... is not grounded in what a skilled artisan would have gleaned from the teachings of the prior art, and is instead an impermissible exercise of hindsight with the claims ... serving as a guide”).

A POSITA would know that in any functioning system of flash management, architectural choices relating the roles of host and controller are not made arbitrarily, and these choices are not made arbitrarily in Ellis or in Son. Ex. 2020 [Jones-Decl.]

9:29-37; Ex. 2026 [U.S. Patent No. 9,183,137] 4:58-65; Ex. 2027 [U.S. Patent No. 9,244,519] 6:41-49; Ex. 2028 [U.S. Patent No. 9,543,025] 6:3-10; Ex. 2029 [U.S. Patent No. 9,671,962] 4:58-65; Ex. 2030 [U.S. Patent No. 10,546,648] 4:49-56.

¶ 39. This background knowledge is also reflected in claims of the '183 patent, which recite functionalities of the claimed “memory controller” and “host,” and specific interchanges of information between them, at length. *See, e.g.*, Ex. 1001 ['183] cl. 1; Ex. 2020 [Jones-Decl.] ¶ 39.

This is not, moreover, a case where a patent challenger’s primary reference is simply silent on details of implementation that are supplied by a secondary reference. *Cf. KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 417 (2007) (showing a motivation to combine may be more difficult in cases that “involve more than the simple substitution of one known element for another or the mere application of a known technique to a piece of prior art ready for the improvement”). Ellis expressly places almost all of the work of flash management—*e.g.*, directing logical read and write operations to specific physical units of memory, managing erase operations, and performing maintenance tasks such as wear leveling and garbage collection—within Ellis’s “controller.” Ex. 2020 [Jones-Decl.] ¶ 40; *see, e.g.*, Ex. 1004 [Ellis] 15:45-50 (describing “logical-to-physical mapping” being “performed by the block control module 306” within Ellis’s memory controller); 4:66-5:3 (locating the “block control module 306” and “memory access module 308” within Ellis’s memory controller); 11:58-62 (locating a “program-erase count” within the “block control module 306” within Ellis’s memory controller); 12:51-55 (locating “recycling operations” within the “memory access module 308” within Ellis’s memory

controller), 14:21-24 (locating “remapping of . . . unusable blocks” within the “memory access module” within Ellis’s memory controller). Son expressly assigns the management of similar tasks to a component called an “SSD module.” Ex. 2020 [Jones-Decl.] ¶ 41; Ex. 1005 [Son] [0039] (locating “a wear count” in Son’s “superblock table 350”), [0033] (describing Son’s superblock table within “management data structures maintained by an SSD module”), [0038] (describing “an FLBA to PLBA mapping table” among the structures maintained by its “SSD module”). Petitioner endeavors to blend aspects of Son’s “SSD module” with Ellis’s host. *See, e.g.*, Pet., 19 (referring to Son’s “SSD module [(host)]”).

A POSITA would recognize that Ellis and Son are not merely alternative ways of allocating responsibility for flash management, but potentially conflicting architectures, representing different series of design choices that are, in any practical system, mutually exclusive. Ex. 2020 [Jones-Decl.] ¶ 42. In particular, Ellis and Son are not complementary: a system that conceals aspects of flash management from its host by handling it within a controller (as in Ellis) generally would not *also* expose those aspects for direct handling by software on a host (as in Son), for example. *Id.* Where references contain distinct and conflicting teachings about the same things, a Petitioner must do more than simply choose between them to provide a motivation to combine. *See, e.g., Samsung Elecs. Co., Ltd. v. Dynamics Inc.*, IPR2020-00499, Paper 63 (Final Written Decision), 33 (Aug. 10, 2021) (“Petitioner

proffers arguments that a person of ordinary skill in the art could have combined the teachings of [the two references], but fails to explain sufficiently *why* they would have chosen the specific features relied upon by Petitioner to read on the claim”) (emphasis in original). Even in the context of a single reference (such as Ellis), 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 at 1370-71. Petitioner does not do that here.

B. Petitioner’s Sinclair Combinations Are Underspecified (Grounds III-IV, All Challenged Claims).

Like Petitioner’s Ellis combinations, *see* Section II.A *supra*, Petitioner’s Sinclair combinations are underspecified. In proposing to combine Sinclair with Son, for example, Petitioner states that “a POSA would incorporate Son’s teaching regarding implementation details for superblock metadata and commands, to the metablock metadata and commands in Sinclair.” Pet., 45. As with Petitioner’s combination of Ellis and Son, it is not clear what features of Son are actually being proposed to be incorporated into Sinclair, let alone why. The challenged claims do not recite abstract “metadata and commands,” of course; they recite a “memory controller,” and a “host,” and—across multiple claim limitations—logic for various kinds of specific exchanges of information between these components, including

- “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions,” *see also* Section IV.A (showing that Petitioner’s Sinclair grounds fail to show this limitation),
- “logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions,” *see also* Section IV.B (showing that Petitioner’s Sinclair grounds fail to show this limitation), and
- “logic to receive an erasure request from the host, via the host interface, the erasure request accompanied by the address corresponding to the specific one of the subdivisions.” *See also* Section IV.C III.C (showing that Petitioner’s Sinclair grounds fail to show this limitation).

To propose to alter Sinclair’s with “details” of “superblock metadata and commands” does not specify (let alone motivate) how the combined system would implement any of these functionalities.

In making arguments about Sinclair in connection with individual claim limitations, moreover, Petitioner repeatedly attempts to incorporate its reasons for combining *Ellis* and Son by reference, as if they could motivate and show how Petitioner’s combination of *Sinclair* and Son would work. *See, e.g.,* Pet., 47 (arguing that “To the extent Sinclair does not teach this limitation, a POSA would be

motivated to incorporate the additional variants of superblock metadata disclosed in Son. As discussed for Ground I, Claim 1[c], Son teaches the claimed page utilization metadata,” where “Ground I” is an Ellis ground). Petitioner repeats this style of argument with Yu, a third reference in both Sinclair combinations. *See, e.g.*, Pet., 50 (arguing that “to apply the teaching of Yu to the combined system [of Sinclair and Son], a POSA would understand that the information sent by the controller to the host in connection with the S.M.A.R.T. command must include a superblock or metablock address, as described for Ground I, Claim 1[d]-[f],” where “Ground I” is an Ellis combination that does not include Sinclair or Yu). As the Board has recognized, however, offering identical rationales to combine disparate collections of references is evidence of hindsight, not a showing of 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. ... [R]epeating 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”).

III. PETITIONER'S ELLIS GROUNDS FAIL (GROUNDS I–II, ALL CHALLENGED CLAIMS).

Petitioner's Ground II challenges dependent claims only, and incorporates Petitioner's Ground I (without modification) for all aspects of the underlying independent claims. *See, e.g.,* Pet., 34 (incorporating Petitioner's challenge to dependent claim 2 in Ground I into Ground II's challenge to claim 6), 37 (incorporating Petitioner's challenge to claim 1 in Ground I into Ground II's challenge to claim 9); 41 (similar for the remaining Ground II challenges). Similarly, all of Petitioner's Ground I challenges incorporate Petitioner's Ground I challenge to claim 1 without modification. As one example, Petitioner's Ground I challenge to independent claim 13 (which is the only other independent claim in the '183 patent) is a table that lists a correspondence between each element of claim 13 and each element of claim 1. Pet., 30; *see also, e.g.,* Pet. 30 (challenging dependent claims 14, 18, and 20 simply by reference to claims 2, 8, and 10), 22, 25, 26 (challenging claims 2, 8, and 10, which depend from claim 1, by incorporating and assuming the validity of Petitioner's challenge to claim 1). For these reasons, this Preliminary Response focuses on Petitioner's Ellis-Son challenge to independent claim 1 in its Ellis-Son Ground I.

A. Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.”

Claim 1² requires a “*memory controller* comprising ... logic to receive a write request from a *host* via the *host interface*, the write request accompanied by an address that designates a specific one of the subdivisions.” Petitioner identifies Ellis’s controller as the claimed “memory controller,” Ellis’s “host interface” as the claimed “host interface,” and Ellis’s “host system” as the claimed “host.” *See, e.g.*, Pet., 14; *see also* Pet., 12-13. Petitioner then offers an argument that Ellis teaches this aspect of the claim, and an argument that Son does. Both arguments fail.

1. Ellis Does Not Teach Or Render Obvious A Write Request “from a host . . . accompanied by an address that designates a specific one of the subdivisions.”

The claims, again, require a memory controller comprising “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.” Petitioner identifies Ellis’s host system as the claimed “host,” and Ellis’s controller as the claimed “memory controller,” and a write request in Ellis as the claimed “write request,” with Ellis’s “super blocks” as the claimed “subdivisions.” Pet., 14-15. For the claimed

² All Ellis challenges depend upon the challenge to claim 1. *See* § III *supra*.

“address that designates a specific one of the subdivisions,” Petitioner identifies “super block identification (ID) values” described in Ellis. *See, e.g.*, Pet. 14-15 (quoting Ex. 1004 [Ellis] 10:27-32). To begin connecting these individual identifications to the claim limitation, Petitioner argues that a “write address [in Ellis] ***necessarily must*** identify or reference a superblock ID value” (*i.e.*, what Petitioner identifies as an “address that designates a specific one of the subdivisions”). Pet., 14-15.³

As a threshold matter, this phrasing does not directly address the claim language, which does not recite a write request ***identifying*** or ***referencing*** a specific one of the subdivisions, but “a write request from a host . . . the write request ***accompanied by*** an address that designates a specific one of the subdivisions.” Petitioner’s cited disclosures do not show any write request from Ellis host system to be ***accompanied by*** any “superblock ID value.” In a portion of Ellis that Petitioner cites but does not quote, for example, Ellis says only that “Actions including host writes, recycling operations, or erasing of blocks can be performed on a set of the erase blocks 408 included within one of the super blocks 502 as a single unit,” Ex. 1004 [Ellis] 8:4-7; *see also* Pet., 14 (citing Ex. 1004 [Ellis] 7:64-8:7). This

³ Following Petitioner, *see, e.g.*, Pet., 7, this Preliminary Response uses “superblock” and “super block” interchangeably with reference to Ellis.

disclosure that actions “can be performed on a set of erase blocks . . . as a single unit” does not say *how* these actions are performed; in particular, it does not say anything about the structure of a write request from Ellis’s host, let alone that such a request is accompanied by a “superblock ID value.” As Dr. Jones explains:

Petitioner’s argument rests on a statement that a “write address” in Ellis “necessarily must identify or reference a superblock ID value,” Pet., 14-15—a contention that Petitioner follows with the bracketed claim language “accompanied by an address that designates a specific one of the subdivisions.” Pet., 14-15.

The argument implied by this juxtaposition—that a POSITA would understand that a write request from the host in Ellis *necessarily must* be “accompanied by” a superblock ID value, because the request may ultimately result in a write operation that identifies or references specific superblocks—is not true. A POSITA would understand, for example, that a write request received by Ellis’s controller from Ellis’s host could be directed to a range of *logical* addresses, with Ellis’s controller tasked with converting logical addresses to physical addresses upon receipt of the request. These logical addresses may not correspond to any physical addresses when the request is received by the controller. In such a situation, any relevant superblock ID values would be identified at the controller only *after* the request was received from the host and physical addresses were assigned by Ellis’s controller as a part of carrying out the request. A POSITA would not understand such a write request to have been “accompanied by” something that Ellis’s controller only generates after it has received that request.

This difference—host write requests contain logical addresses, not superblock IDs—is perhaps obscured by Petitioner’s ambiguous phrase ‘write address,’ Pet., 14—a phrase that invites confusion between logical and physical addresses, and hence confusion between what the alleged “write request from [the] host” in Ellis is “accompanied by,” on the one hand, and what Ellis’s controller might later create or identify, on the other.

Ex. 2020 [Jones-Decl.] ¶¶ 49-51.

Ellis is replete with suggestions that its “host system” would be entirely unaware of what “superblock ID values” are associated with the data it is storing, let alone any “superblock ID values” associated with any write request. As one example, Ellis expressly refers to “superblock ID values” (among other examples of “metadata” maintained in Ellis’s system) as “non-host related information.” Ex. 1004 [Ellis] 10:21-32. As another example, Ellis describes its “metadata” (which includes “superblock ID values”) as stored on its controller, and as managed by the “block control module 306,” which is a component of Ellis’s controller. *Id.*, 10:3-20, 4:67-5:3. As another example, Ellis describes a “logical-to-physical (L2P) tabling system within the runtime volatile memory” of its controller, *id.*, 12:3-14, 3:39-42, 4:4-8 (locating volatile memory within Ellis’s controller), and “logical-to-physical mapping of ... host data within the memory devices 110” as “non-host related information” stored on its controller. *Id.* 10:25-27. As another example, Ellis

states, *specifically in connection with a host sending a write request to its controller*, that “The host system 108 can request the memory controller 104 for reading, writing, and deleting data from or to *a logical address space* of a storage device or the memory sub-system 102 that includes the memory array 106,” Ex. 1004 [Ellis] 3:25-28. As Dr. Jones explains:

From these and many other disclosures in Ellis, a POSITA would understand write requests originating in Ellis’s host would specify, at most, logical addresses that indicate locations in the host’s logical address space (*e.g.*, addresses corresponding to a file-system or OS-level views of storage, not physical locations in flash memory). A POSITA would understand Ellis’s super block “ID values,” by contrast, specify physical locations in flash memory. To the extent that such “ID values” are used at all in handling a write request in Ellis’s system, they would be generated at Ellis’s controller. Ellis’s controller is the component in Ellis’s system tasked with translating logical addresses to physical ones, and with assigning physical addresses for logical addresses that have not previously been written to.

Ex. 2020 [Jones-Decl.] ¶ 53. For at least these reasons, Ellis does not teach the claimed “write request.”

2. Son Does Not Teach Or Render Obvious A Write Request “from a host . . . accompanied by an address that designates a specific one of the subdivisions” In Petitioner’s Combination.

Petitioner attempts to save Ellis’s failure to teach a write request “*from a host* ... accompanied by an address that designates a specific one of the subdivisions” by

arguing that “[t]o the extent Ellis does not teach this limitation, a POSA would be motivated to combine its teaching with Son.” Pet., 15. Petitioner points to “write requests” in Son’s system, arguing that “write requests [*in Son*] include an assigned file system logical block address (‘FLBA’) which is associated with a ‘superblock address.’” *Id.* (quoting Son). But Petitioner does not explain how write requests *in Son’s system* relate to write requests received by the “memory controller” in its combination—a modified version of the controller of Ellis—or why Son’s FLBA being “associated with” a “superblock address” in Son teaches that a write request would be “accompanied by” a superblock ID value in the combined system.

Indeed, while Petitioner proposes to “combine Son’s host functionality with the system described in Ellis,” Pet., 10, Petitioner is vague about how write requests actually work in the combined system—except to suggest, for example, that they “incorporate the additional forms of superblock metadata described in Son.” Pet., 10. As has already been explained, write requests from Ellis’s host to Ellis’s (unmodified) controller are not accompanied by “super block ID values,” *see* Section III.A.1 *supra*, so presumably Petitioner is proposing that these “additional forms of superblock metadata” would include such information. But Petitioner also says that “Son teaches a ‘write request’ accompanied by an FLBA designating a superblock address.” Pet., 15. But Son’s “FLBAs” only “designate” superblock addresses in the sense that “mapping tables” at Son’s SSD module associate them with superblock

addresses, Ex. 2020 [Jones-Decl.] ¶ 57; Ex. 1005 [Son], ¶¶ [0038] (noting that a “mapping table” in Son’s Figure 3B “associates each FLBA ... with a super block address”), [0034] (locating the mapping tables at Son’s “SSD module”)—and Petitioner does not argue or show that these mapping tables are on the host in its combination. As Dr. Jones observes:

Petitioner’s analogy to Son’s FLBAs thus presents the same problem as Petitioner’s argument about Ellis. While Petitioner describes the combination of Ellis and Son at a high level, in terms of adding “additional forms of superblock metadata” and “the sharing of superblock metadata with a host” to Ellis, *see* Pet., 9-11, this does not say that Son’s “mapping tables” (or Ellis’s logical-to-physical mapping functionality) would exist on the host of the combined system. In the absence of such a modification, in the combined system, a write request from the host would presumably still be resolved to a “superblock ID value” at the controller, as it is in Ellis.

If Petitioner is proposing to augment the host in the combined system with the logical-to-physical mapping *function* of Son’s SSD module—a proposal that is far from clear in Petitioner’s vague reference to adding “additional forms of superblock *metadata* described in Son” to the host system of Ellis, Pet., 10—then Petitioner does not explain how or whether the memory controller in Ellis, which *also* maintains logical-to-physical mapping information (and which Ellis also refers to as “metadata,” Ex. 1004 [Ellis] 10:21-27), would be modified as to be compatible with this new host functionality.

Ex. 2020 [Jones-Decl.] ¶¶ 58-59. As Dr. Jones further explains, it is not clear to what extent the host of the combined system would be used to replace functionality that is present in Ellis’s unmodified controller:

It does not appear, for example, that Petitioner proposes that Ellis’s controller plays *no* role in managing Son’s “superblock metadata” (whether Ellis’s or Son’s) in Petitioner’s combination. In connection with other claim limitations, for example, Petitioner argues that because “in Ellis, the superblock metadata is updated by the controller,” that “[i]n the combined system *the additional superblock metadata described in Son* would also be updated by the controller.” Pet., 15. Yet in Son, it is Son’s SSD module—not Son’s memory controller, and not a component that Petitioner analogizes to any controller in the combination—that is tasked with updating Son’s superblock-level metadata. *See, e.g.*, Ex. 1005 [Son] [0050]-[0055] (describing Son’s “SSD module” as updating various mapping tables, including how to “associate the FLBA [of a write operation] with [a] newly selected PLBA,” during a write operation). Petitioner does not indicate how these distinct approaches would be reconciled.

In any event, a POSITA would recognize a change of this sort to represent a fundamental change in the operating principles of the system—with implications far beyond simply what “metadata” accompanies a write request. As one example, a POSITA would recognize that if a host and a memory controller were *both* maintaining logical-to-physical addressing information for the same memory, the system would need some way of synchronizing their activity (*e.g.*, to

avoid the host and controller potentially operating upon different logical-to-physical mappings for the same memory). Designing, specifying, and validating that mechanism would be a substantial undertaking, potentially impacting not just the “metadata” exchanged between components, but the command set used by the host, on-media data structures in both components, and scheduling policies in both components. A POSITA would not view such a hybrid system as a minor variant of a host-centric system like Son or a controller-centric system like Ellis.

Ex. 2020 [Jones-Decl.] ¶¶ 60-61.

For at least these reasons, Petitioner does not show how a write request “accompanied by an address that designates a specific one of the subdivisions” is taught in its combination. *See Johns Manville Corp. v. Knauf Insulation, Inc.*, IPR2018-00827, Paper 9, 11-12 (Oct. 16, 2018) (informative) (denying institution when “Petitioner does not compare the thermoset binder disclosed in Worthington with the thermoset binder disclosed in Srinivasan, nor does Petitioner attempt to show that one of ordinary skill in the art would have had a reason to substitute Worthington’s binder for the binder disclosed in Srinivasan”).

B. Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.”

Claim 1 additionally requires “logic to send the host the information

indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.”⁴

Petitioner’s identifies Ellis’s “host system” as the claimed “host” and asserts that Ellis’s disclosure of “control or status information” sent from the controller to the host encompasses the sending of superblock-level metadata, including valid page counts. *See, e.g.*, Pet., 14 (citing Ex. 1004 [Ellis] 4:46-50 in identifying the “host interface” in Ellis with the claimed host interface, and the “host system” in Ellis with the claimed “host”), *id.* (citing Ex. 1004 [Ellis] 14:25-34 for a “write operation . . . for host write data from the host system 108” in connection with the claimed “write request from a host”); *id.*, 16 (arguing that Ellis discloses sending “valid page count” statistics for a single “super block” to Ellis’s “host system”).

But Ellis *never* discloses transmitting superblock-specific metadata—let alone valid-page counts—to its host system; Ellis describes this type of metadata as “non-host related.” *See* Section III.B.1 *infra*. Petitioner attempts to supplement these defects of Ellis with citations to Son—suggesting that Son’s “SSD module” demonstrates how page utilization statistics might be communicated to a host. In doing so, however, Petitioner appears to assume without justification that Son’s

⁴ Again, all of Petitioner’s Ellis grounds rely upon Petitioner’s arguments involving Ellis and Son against independent claim 1. *See* Section III *supra*.

“SSD module” would be regarded by a POSITA as interchangeable with Ellis’s host system. Petitioner further ignores that in Son, Son’s “SSD module” *itself* generates and updates the information that Petitioner says a host would request from a controller in its combination. Ex. 2020 [Jones-Decl.] ¶ 65. Petitioner’s failure to explain these issues is fatal to Petitioner’s combination. *See* Section III.B.2 *infra*.

1. Ellis Does Not Teach Sending “information indicating extent of page utilization of [a] specific one of [its] subdivisions” To The Alleged “host” In Petitioner’s Combinations.

Ellis discloses that its controller may send “control or status information generated by the memory controller 104 to the host system 108,” Ex. 1004 [Ellis] 4:51-53, and Petitioner cites this disclosure for the proposition that “Ellis therefore teaches ‘logic to send the host [] information.’” Pet., 16. To argue that Ellis teaches information “*indicating extent of page utilization* of [a] specific one of the subdivisions,” Petitioner points to “NAND statistics” maintained for individual “super blocks” in Ellis by Ellis’s memory controller. Pet., 16.

Ellis’s memory controller is not what Petitioner identifies as the “host.” Critically, Ellis does not disclose that these statistics are shared with Ellis’s host system—Petitioner’s alleged “host”—and Ellis does not disclose any reason why its memory controller would share these statistics with its host system. Petitioner attempts to overcome this lack of disclosure by relying upon its expert for its inference that Ellis’s “control and status information,” which Ellis *does* describe as

sent to its host system, somehow includes NAND statistics. Pet., 16. But this is pure *ipse dixit*.

Ellis ***nowhere*** discloses any transmission of “super block”-level information from Ellis’s memory controller to Ellis’s “host system,” whether as part of Ellis’s “control and status information” (as Petitioner argues, Pet., 16), or as part of any other transmission from Ellis’s controller to Ellis’s host system. In particular, Ellis nowhere discloses sending any “valid page count” statistics for a single “super block” to Ellis’s “host system” (as Petitioner argues, Pet., 16). A POSITA would also recognize that Ellis nowhere discloses any action taken by its host system that would require or even advantageously be able to use statistics such as “valid page count” statistics for a single “super block.” Ex. 2020 [Jones-Decl.] ¶ 70.

Petitioner says that “Ellis teaches that the control and status information”—specifically, the “control and status information” that Ellis describes its controller sending to its host system at 4:51-53—“is stored as metadata,” citing disclosures in Ellis about “metadata.” Pet., 16. Equating Ellis’s “control and status information” with Ellis’s “metadata” in this fashion is the product of Petitioner’s own invention, not a disclosure or teaching of Ellis. Ellis nowhere expressly refers to the “control and status information” it describes at 4:51-53 as “metadata,” or suggests that its “control and status information” includes “metadata.” To the contrary, Ellis expressly and repeatedly describes “metadata” as “***non-host related*** information,”

see, e.g., Ex. 1004 [Ellis] 10:21, 10:25, 10:28, and a POSITA would recognize from these disclosures that Ellis’s “metadata”—a category that includes the “NAND statistics” identified by Petitioner, Pet., 16—is ***not*** sent to Ellis’s “host system.” Ex. 2020 [Jones-Decl.] ¶ 71.

As another example, where Ellis describes “logical-to-physical mapping information of the host data within the memory devices 110” as an example of “metadata,” a POSITA would understand that this information is ***exclusively*** maintained by the controller in Ellis, and is not sent to Ellis’s “host system.” Ex. 2020 [Jones-Decl.] ¶ 72. Ellis’s other examples of “metadata,” such as “logical block address (LBA) information, [and] super block identification (ID) values” cited by Petitioner (Pet., 16-17), are also not sent to Ellis’s “host system.” *Id.*

Petitioner attempts to circumvent Ellis’s lack of relevant disclosure by fiat—by referring, without justification, to these examples as “stored ‘control and status’ metadata ***sent to the host.***” Pet., 16 (citing Ex. 1004 [Ellis] 10:27-32, 12:26-41). But the cited portions of Ellis only show that Ellis regards LBA information and super block identification values as examples of “metadata”—not that these examples are “control and status information” in the sense of Ellis 4:51-53, and certainly not that these examples are ever “sent to the host” of Ellis.

Indeed, although Petitioner states that “Ellis describes instances where the controller sends the host ‘metadata’ showing the ‘valid state of data stored within the

one of the superblocks,” Pet., 16 (quoting Ex. 1004 [Ellis] 12:34-37). Ellis discloses no such thing. The portion of Ellis cited by Petitioner states in full:

During portions of drive operations such as but not limited to drive initialization, the metadata pages 508 can be used to restore a host logical to physical mapping of data and a valid state of data stored within the one of the super blocks 502.

Ex. 1004 [Ellis] 12:34-37. As Dr. Jones observes:

This disclosure does not teach sending Ellis’s “metadata” to Ellis’s “host.” A POSITA would understand Ellis’s reference to the use of the “metadata pages 508” to “restore a host logical to physical mapping” to refer to operations that take place entirely within Ellis’s controller, where Ellis stores its logical-to-physical mapping information. Ex. 1004 [Ellis] 10:24-27 (referring to “logical-to-physical mapping information of the host data within the memory devices” as “non-host related information”). Ellis discloses no reason why updating or “restor[ing]” this information would involve, let alone require, sending any of this information to Ellis’s host.

Ex. 2020 [Jones-Decl.] ¶ 75.

Given the limited responsibilities of Ellis’s “host system,” a POSITA would understand Ellis’s “control and status information” to include summary information appropriate for such a “host system,” potentially including high-level endurance statistics (e.g. estimated life left computed from program/erase cycle totals or

averages), high-level error statistics (e.g. read error counts), temperature alerts, and the like. Ex. 2020 [Jones-Decl.] ¶ 76. As Dr. Jones explains:

What such exemplary types of information have in common is that they relate to the “control and status” of the *drive as a whole*, not to the management of individual “super blocks.” At the time of the invention, the exposure of drive-level summary information to the host was common in systems that, like Ellis, delegated most of the responsibility for flash management to the memory controller. But there would be no reason for a memory controller in such a system to report information at finer levels of granularity to the host, because the host in such a system—not being tasked with flash management—has no use for such information.

Ex. 2020 [Jones-Decl.] ¶ 77. In sum, Ellis does not teach the claimed “logic to send the host the information indicating extent of page utilization.”

2. Son Does Not Teach Or Render Obvious Sending “information indicating extent of page utilization of [a] specific one of [its] subdivisions” To The Alleged “host” In Petitioner’s Combinations.

As noted above, the alleged “host” in Petitioner’s combination is the “host system” in Ellis. As alleged support for sending the claimed “information indicating extent of page utilization” in its combination, Petitioner identifies disclosures in Son suggesting that Son’s “SSD module” may query Son’s “controller” for information such as “the number of free, valid and/or invalid pages in each candidate superblock” in Son. *See, e.g.*, Pet., 17-18 (quoting, e.g., Ex.1005 [Son] [0047]). Here Petitioner

at times appears to replace the “host system” of Ellis (formerly the alleged “host” of claim 1) with the “SSD module” of Son, or at least to invite an inference that any disclosures regarding Ellis’s “host system” are interchangeable with disclosures regarding Son’s “SSD module,” by describing disclosures about Son’s “SSD module” as if they are teachings about the claimed “host.” *See, e.g.*, Pet., 17 (“Son teaches that the SSD is managed by a host which is comprised of an ‘SSD module’”), *Id.* (“Son teaches that the host may acquire superblock metadata stored in flash memory”), 18 (“The host may query page utilization”).

Petitioner never squarely addresses this issue, or offers evidence in support of the implicit argument that a POSITA would recognize capabilities of Son’s “SSD module” to be interchangeable with those of Ellis’s “host system.” And a POSITA would not recognize these components of Ellis and Son to be interchangeable. As one example, the only reason that Petitioner identifies for Son’s “SSD module” to query any alleged “information indicating the extent of page utilization” in Son is “in order to perform operations such as ‘garbage collection and wear leveling’” in Son. Pet., 18 (quoting Ex. 1005 [Son] [0017]). As Petitioner recognizes, garbage collection and wear leveling in Son are performed by Son’s “SSD module.” Petitioner does not address the fact that in *Ellis*, garbage collection and wear leveling to be performed by Ellis’s *controller*, not Ellis’s “host system.” Ex. 2020 [Jones-Decl.] ¶ 81. As Dr. Jones explains:

A POSITA would generally recognize that a flash memory system such as Ellis would perform garbage collection and wear leveling—a point that Petitioner’s expert would appear to acknowledge, *see, e.g.*, Ex. 1003 [Litz-Decl.] ¶ 33 (describing garbage collection in section entitled “Technology Background and State of the Art” as a feature of managing solid-state drives), 35 (in the same section, describing wear leveling as “essential” in flash management). And Ellis expressly discloses, for example, that its controller keeps track of numbers of valid pages, program/erase cycles, and the like—inputs that a POSITA would understand to be important to garbage collection or wear leveling (again, as Petitioner’s expert generally acknowledges; *see* Ex. 1003 [Litz-Decl.] ¶¶ 33-34 (describing the general role of “valid blocks” in garbage collection), 35 (discussing the limited number of P/E cycles in connection with wear leveling)).

Although Ellis does not employ the term “garbage collection,” Ellis expressly describes its “memory access module” (which resides within Ellis’s controller, Ex. 1004 [Ellis] 4:67-5:3) as performing “recycle operations” for reasons “including freeing space and wear leveling,” Ex. 1004 [Ellis] 16:4-13), and a POSITA would recognize that garbage collection involves “freeing space.” Because Ellis never discloses any role for its host system in performing such operations (or in accessing or updating any of the relevant inputs), a POSITA would understand that flash garbage collection and wear leveling operations in Ellis are performed exclusively by Ellis’s controller, without involvement of Ellis’s host.

Ex. 2020 [Jones-Decl.] ¶¶ 82-83.

Thus, the very information that Son’s SSD module uses and requires for garbage collection and wear leveling—the supposed inspiration for sending “information indicating extent of page utilization” to the host in the combination—is information that Ellis’s “host system” plays no role in receiving or using. Ex. 2020 [Jones-Decl.] ¶ 84. Petitioner nowhere addresses this.

Petitioner nowhere expressly proposes, for example, that the modified Ellis host in its combination would take over garbage collection and wear leveling operations in the combination. Petitioner states only at a high level that Son teaches “the benefit of *sharing information* with the host in order to perform memory management functions,” Pet., 17. As Dr. Jones further explains:

It is unclear whether Petitioner suggests that the modified host would perform garbage collection and wear leveling. This type of modification is arguably hinted at by Petitioner’s argument that Son’s use of data relevant to garbage collection and wear leveling teaches “send[ing] the host the information indicating extent of page utilization” in the combined system. But Petitioner does not expressly identify this (or anything else) as a reason for sending this information to the host in the combined system.

If Petitioner were to propose that Ellis’s modified host either share or completely take over garbage collection or wear leveling responsibilities in the combination, this would not be a minor modification of Ellis. Such a modification would go far beyond simply what types of “metadata” exist in Ellis’s system. A POSITA would

recognize such an expansion of the role of Ellis’s host to raise other unanswered questions of system design. Petitioner does not address, for example, what would happen with garbage collection and wear leveling *as already performed in Ellis’s controller* in the combination. (If Petitioner were proposing that the modified Ellis host would exclusively manage these operations, for example, Petitioner does not explain why Ellis’s controller would still be tasked with managing and updating the inputs to these operations, if its only role in these operations is to send that information to the host.)

Ex. 2020 [Jones-Decl.] ¶¶ 86-87.

For at least these reasons, Petitioner does not show that its combination teaches sending the host the claimed “information indicating extent of page utilization.”

C. Petitioner Does Not Show That Its Ellis Combinations Teach Or Render Obvious “an erasure request from the host . . . the erasure request accompanied by the address corresponding to the specific one of the subdivisions” (Grounds I-II, All Challenged Claims).

Claim 1 additionally requires “an erasure request from the host . . . the erasure request accompanied by the address corresponding to the specific one of the subdivisions.”⁵ But Petitioner fails to demonstrate that (1) Ellis discloses host-

⁵ Again, Petitioner’s Ellis grounds all rely upon Petitioner’s arguments involving Ellis and Son against independent claim 1. *See* Section III *supra*.

initiated erase requests, (2) why any host-initiated erase request assumed to exist would be “accompanied by” what Petitioner identifies as the claimed “address,” or (3) that Son would teach this limitation in the combination if Ellis alone does not.

1. Petitioner Does Not Show Erasure Requests From Ellis’s Host.

As an initial matter, Petitioner does not show how Ellis discloses “an erasure request *from the host*” Ellis discloses that sets of erase blocks in “super blocks” of its system may be managed “as a single unit.” *See, e.g.*, Ex. 1004 [Ellis] 8:4-7 (“Actions including host writes, recycling operations, or erasing of blocks can be performed on a set of the erase blocks 408 included within one of the super blocks 502 as a single unit”). This does not say that Ellis’s *host* issues requests to erase specific blocks or super blocks. Pointing to this disclosure, however, Petitioner argues that “Ellis teaches that *the host or controller* may request ‘erasing of blocks ... within one of the super blocks 502 as a single unit.’” Pet., 18 (quoting Ex. 1004 [Ellis] 8:2-7 and citing Ex. 1004 [Ellis] 4:58-65).

Petitioner’s use of the disjunctive—“the host or controller”—obscures the fact that Petitioner never points to *any* teaching indicating that Ellis’s host system initiates erase requests, let alone such a request “accompanied by” what Petitioner identifies as an “address corresponding to the specific one of the subdivisions.” For the “host” aspect of this assertion, Petitioner relies upon Ellis’s column 4 disclosure (discussed in detail in Section **Error! Reference source not found.**) to interpret

Ellis's disclosures about its controller as if they are teachings about its host. Pet., 18.

Again, this high-level disclosure states generally “that the memory controller 104, the host system 108, and the memory array 106 can operate any of the modules and functions of the memory controller 104.” Ex. 1004 [Ellis] 4:58-65. This does not say that Ellis's host sends erase requests to Ellis's controller. And a high-level disclosure is not a teaching that a POSITA would be motivated to realize every alternative that might be included in a high-level disclosure. *See, e.g., Vicor Corp. v. Delta Elecs., Inc.*, IPR2024-00715, Paper 13, 21 (Oct. 2, 2024) (denying institution) (“[W]hile [a reference's] ‘disclosures indicate that some different types of contacts could be used, the issue remains as to why a skilled artisan would have selected [the specific contact] of [a secondary reference] for the modification, as well as to modify [a reference] in the manner that Petitioner proposes.’”); *Peddinghaus Corp. v. Ficep Corp.*, IPR2020-01707, Paper 15, 37 (May 11, 2021) (denying institution) (finding obviousness argument “deficient[] for selectively picking and choosing from [a] second embodiment only so much of it to support the alleged unpatentability, to the exclusion of other features that would undermine the assertion of unpatentability”).

Separately from citing Ellis's column 4, Petitioner asserts that “Ellis teaches host erase requests may be sent, via the ‘host interface,’ to the ‘memory controller,’ which has ‘logic’ to process those commands.” Pet., 18. But again, Petitioner does

not identify any host-initiated erase operations in Ellis. Petitioner suggests that it has done so by introducing the phrase “host erase request,” but this term does not appear in Ellis. The cited disclosures concern only general aspects of Ellis’s system—not anything specific to erase requests. *Compare* Pet., 18 (citing Ex. 1004 [Ellis] 4:46-50, 3:38-45, 20:40-44) *with* Ex. 1004 [Ellis] 4:46-50 (“The host interface unit 208 allows the host system 108 of FIG. 1 to interface and interact with the memory controller 104. The host interface unit 208 can include a host interface 218 to provide communication mechanism between the host interface unit 208 and the host system 108”), 3:38-45 (describing Ellis’s Figure 2, an “exemplary hardware block diagram of the memory controller 104”), 20:40-44 (“Functions or operations of the memory controller 104 as described above can be implemented with the control unit 202 of FIG. 2, the storage unit 204 of FIG. 2, the memory interface unit 206 of FIG. 2, the host interface unit 208 of FIG. 2, or a combination thereof”⁶). Such high-level disclosures do not show the claim limitation.

⁶ Each of the components 202, 204, 206, and 208 of Figure 2 is contained within Ellis’s memory controller. *See, e.g.*, Ex. 1004 [Ellis] 3:38-39 (“Referring now to FIG. 2, therein is shown an exemplary hardware block diagram of the memory controller 104.”)

2. Petitioner Does Not Show That A Host-Initiated Erase Request Assumed To Exist In Ellis Would Be “accompanied by [an] address corresponding to the specific one of the subdivisions.”

Even assuming, *arguendo*, that Petitioner has shown that Ellis teaches host-initiated erase requests (and Petitioner has not), Petitioner does not show that any alleged erase request coming from Ellis’s host system would be “*accompanied by* the address corresponding to the specific one of the subdivisions” (which Petitioner identifies as a “super block ID value” in Ellis). Pet., 19 (emphasis in original).

Although Petitioner does not use the term “inherency,” its argument is expressly one of inherency: Petitioner states that “[a] POSITA would understand that, for erasures to be carried out on a superblock as a single unit, the erasure request *must* be accompanied by an address corresponding to ‘superblock ID value.’” Pet., 18-19. *See, e.g., Trintec Indus., Inc. v. Top U.S.A. Corp.*, 295 F.3d 1292, 1295 (Fed. Cir. 2002) (inherency requires demonstrating that a limitation is “‘necessarily present,’ not merely probably or possibly present, in the prior art.”); *see also PAR Pharm. v. TWi Pharms., Inc.*, 773 F.3d 1186, 1195–96 (Fed. Cir. 2014) (similar); *Southwire Co. v. Cerro Wire LLC*, 870 F.3d 1306, 1310–11 (Fed. Cir. 2017) (similar).

But Petitioner does not remotely show that any such erase request would “necessarily” be accompanied by the super block ID value. Whenever Ellis expressly discloses its “host system” requesting operations, Ellis speaks in terms of its host’s logical addressing, *see, e.g.,* Ex. 1004 [Ellis] 3:25-28 (“The host system

108 can request the memory controller 104 for reading, writing, and deleting data from or to *a logical address space* of a storage device or the memory sub-system 102 that includes the memory array 106”); *see also id.*, 5:21-23 (“The term ‘logical’ refers to *non-physical* and thus does not represent the physical configuration of the memory devices 110”), *cf. id.*, 7:65-66 (“The super blocks 502 are defined as groups of the erase blocks 408”), while it only speaks of “super block identification (ID) values,” as resident in data structures stored on and maintained by Ellis’s controller. *See id.*, 10:21-32. As Dr. Jones explains:

A POSITA would understand that host requests in Ellis could have the ultimate effect of “super blocks” being erased as single units without the request itself being accompanied by any “superblock ID value” from Ellis’s host. A POSITA would recognize, for example, that if Ellis’s host were to send a hypothetical command to erase an entire drive, or to erase the contents of a given range of a logical addresses, such a request would have the ultimate effect of erasing superblocks even though it could be carried out without being accompanied by the superblock ID value of any superblock implicated by the request. A POSITA would understand Ellis’s controller to be capable of generating the superblock ID values relevant to such a request without the host providing them.

Ex. 2020 [Jones-Decl.] ¶ 99.

Indeed, and perhaps to the contrary, Ellis refers to its super block ID values as

“non-host related information”—that is, information generated and exclusively used internally by its controller. *See, e.g.*, Ex. 1004 [Ellis] 10:24-30; Ex. 2020 [Jones-Decl.] ¶ 100.

At most, Petitioner speculates that because an erase operation ultimately acts on superblocks, it could be accompanied by an express identification of affected superblocks in terms of their superblock ID values. But inherency is not established by speculating that a limitation *could* be present; the limitation must be shown to be necessarily present. Petitioner does not make any such showing.

3. Son Does Not Teach Or Render Obvious The Claimed Erasure Request In Petitioner’s Combination.

In tacit acknowledgment of Ellis’s failure to teach the claimed “logic to receive an erasure request from a host,” Petitioner offers an alternative argument: that Son provides what Petitioner terms “additional implementation details for host initiated erase commands.” Pet., 19. Petitioner’s arguments concern disclosures in Son relating to the capabilities of Son’s “SSD module.” Pet., 19-20.

As already noted in connection with other limitations above, however, Petitioner does not show why a POSITA would regard every capability of Son’s “SSD module” as a capability of Ellis’s “host system”; Petitioner is simply inviting the unwarranted inference that Son’s “SSD module” can fulfill any role in its combination that Ellis’s “host system” cannot. *See, e.g.*, Pet., 19 (stating that in Son,

“the SSD module [(host)] initiates the deletion operation,” quoting Son at [0059]) (emphasis and parenthetical assertion in original).

Petitioner also does not explain why a host-initiated erase operation inspired by Son would be “accompanied by” any what it appears to identify as an “address corresponding to ... [a] ... subdivision” in Son. Petitioner states that in Son, “the SSD module [(host)] initiates [a] deletion operation’ by issuing “an ‘explicit command ... in association with a particular FLBA,” to the controller,” and that “[t]he delete command [in Son] with FLBA sent to the controller designates a ‘superblock address’ and a ‘PLBA,’” Pet., 19 (quoting Son), where Petitioner sets the term “superblock address” in boldface. But as already discussed, however, Son’s “FLBAs” only resolve to superblock addresses in Son in the sense that tables in Son’s host-side SSD module associate them with superblock addresses. *See* Section III.A.2. In Ellis, the analogous functionality of logical-to-physical address translation occurs *at the controller*. Ex. 2020 [Jones-Decl.] ¶ 104. Without further specification of how Petitioner’s combination is supposed to operate, *see* Section II.A, it is entirely unclear how a host-initiated erase request inspired by Son’s “FLBAs” could be “accompanied by” a super block ID value. If Petitioner is proposing that *both* the host and the controller of the combination perform logical-to-physical address translation, or that *only* the host of its combination would do so, it never says this, let alone explain why a POSITA would be motivated to provide a

combination of Ellis and Son with that functionality. Accordingly, Petitioner has not shown this limitation to be taught by its combination.

IV. PETITIONER’S SINCLAIR GROUNDS FAIL (GROUNDS III–IV, ALL CHALLENGED CLAIMS)

Petitioner’s Sinclair-based challenges rest on Petitioner’s analysis of claim 1 in view of its combination of Sinclair, Yu, and Son (“Ground III”). *See, e.g.*, Pet., 62, 66, 69 (Petitioner’s Ground IV challenges, to dependent claims only, incorporating Petitioner’s Ground III challenges to independent claims), 52, 55 (referring Petitioner’s Ground III arguments regarding dependent claims to Petitioner’s arguments regarding independent claim 1).

A. Petitioner Does Not Show That Its Sinclair Combinations Render Obvious “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions” (Grounds III-IV, All Challenged Claims).

Claim 1, again, requires a “memory controller” with “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.” Petitioner identifies the claimed “memory controller” with the controller within *e.g.* the “memory system 2” of Sinclair’s Figure 1 (which Sinclair also calls the “memory”), and the claimed “host” with the “host system 1” in Sinclair’s Figure 1 (which Sinclair also calls its

“host”).⁷ See Pet. 46; see also Ex. 1007 [Sinclair], Figure 1, 6:54 (referring to the “memory 2” of Figure 1), 7:25-37 (referring to the “host 1” of Figure 1). Petitioner identifies write commands sent from Sinclair’s host to Sinclair’s controller as the claimed “write requests,” see Pet., 47, and Sinclair’s “metablocks” as the claimed “subdivisions.” See, e.g., Pet., 46.

1. Sinclair Does Not Teach The Claimed “write request from a host . . . accompanied by an address that designates a specific one of the subdivisions.”

With these identifications, Petitioner never expressly shows how what Petitioner identifies as the claimed “write request from a host” in Sinclair is “accompanied by an address that designates a specific one of the subdivisions” as required by claim 1. Petitioner states that “Sinclair teaches that [its] controller has logic to receive host write commands accompanied by ‘*logical addresses assigned by the host*,’” Pet., 47-48 (citing Ex. 1007 [Sinclair] 13:9-12, 12:20-30), and that “[t]he logical addresses accompanying the write request *are converted by the controller* to ‘physical addresses within a metablock,’ e.g., a metablock address designating a specific one of the subdivisions.” Pet., 48 (quoting Ex. 1007 [Sinclair]

⁷ Although Sinclair’s Figure 1 is labeled “(Prior Art),” Sinclair expressly states that Figure 1 is “such a system that the various aspects of the present invention may be implemented.” Ex. 1007 [Sinclair] 6:49-51.

13:11-12). Here Petitioner expressly acknowledges that in the context of the “write requests” it identifies in Sinclair, it is Sinclair’s **controller**, not Sinclair’s **host**, that generates what it identifies as the claimed “address designating a specific one of the subdivisions” (what Petitioner calls a “physical address[] within a metablock,” or “metablock address”). Pet., 48 (citing Ex. 1007 [Sinclair], 13:4-12, 12:20-30).

Thus, even *as Petitioner describes Sinclair*, a write command received by Sinclair’s controller from Sinclair’s host is **not** “*accompanied by*” what Petitioner identifies as an “address that designates a specific one of the subdivisions.” *See, e.g.*, Pet., 46 (identifying “subdivisions” in claim 1 with “metablocks” in Sinclair). What Petitioner terms the “metablock address designating a specific one of the subdivisions” (Pet., 48) is generated at Sinclair’s controller. This does not match the claim. Petitioner’s conclusory assertion that this aspect of Sinclair **does** teach this limitation is pure attorney argument:

Thus, Sinclair teaches that the controller has “logic to receive a write request from a host via the host interface, the write request accompanied by an address that designates a specific one of the subdivisions.”

Pet. 48 (citing nothing). This does not satisfy Petitioner’s burden: “[a]ttorney argument is not evidence.” *See, e.g., Icon Health & Fitness, Inc. v. Strava, Inc.*, 849 F.3d 1034, 1043 (Fed. Cir. 2017) (vacating finding of obviousness); *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. The petitioner must instead articulate specific reasoning, based on evidence of record, to support the legal conclusion of obviousness.”); *see also, e.g., Vectair Sys., Inc. v. Fresh Prods., Inc.*, IPR2024-00824, Paper 9, 19 (Nov. 12, 2024) (denying institution when challenge included “attorney argument with no cited evidentiary support,” and “Petitioner’s primary argument regarding key elements of claim 1 lack[s] adequate evidentiary support”).

In a portion of Petitioner’s expert declaration that Petitioner cites (but does *not* quote) several paragraphs later, Pet., 48 (citing Ex. 1003 [Litz-Decl.] ¶¶ 198-202), Petitioner’s expert offers a slightly different formulation of Petitioner’s argument:

Sinclair teaches that the host issues write requests with an address that designates a metablock: “When the host writes data to the memory system, the controller converts the logical addresses assigned by the host to physical addresses within a metablock.” [Sinclair] 13:9-15. Therefore, a POSA would understand that Sinclair discloses a controller which may receive a write request from the host, where that host includes a logical address; and a POSA would further understand that the logical address—upon conversion to a physical address—designates a metablock.

Ex. 1003 [Litz-Decl.] ¶ 199. This testimony should be disregarded for several

reasons.

First, to any extent that Petitioner’s expert is arguing that a POSITA would understand that a logical address in Sinclair itself “designates” a metablock in Sinclair, such that the claimed “address that designates a specific one of the subdivisions” can be *literally identified* with a *logical address* in a write command from the host in Sinclair—and it is not clear that Petitioner’s expert is even taking this position—this is not an argument that Petitioner makes. *Cf.* Pet., 48 (referring to a “*metablock* address designating a specific one of the subdivisions,” and not to any logical address “designating” any specific one of the subdivisions). Arguments cannot be incorporated by reference. 37 C.F.R. § 42.6(a)(3).

Second, this testimony does not identify any factual support for Petitioner’s argument; it only frames Petitioner’s conclusory statements as statements about what “a POSITA would understand.” This cannot satisfy Petitioner’s burden to show that a limitation is present in the prior art. *See, e.g.*, 37 C.F.R. § 42.65(a); *see also Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9, 15-17 (Aug. 24, 2022) (precedential) (denying institution when expert declaration “adds little to the conclusory assertion for which it is offered to support”).

Third and finally, Petitioner’s expert’s position is at odds with the understanding of the knowledge of the POSITA, as Dr. Jones explains:

Even taking Petitioner’s identification of the claimed “subdivisions” with Sinclair’s superblocks for granted for purposes of argument, to any extent that Petitioner’s expert Dr. Litz is taking the position that write commands in Sinclair are “accompanied by” Petitioner’s alleged “address that designates a specific one of the subdivisions” in the sense of claim 1, I disagree.

Dr. Litz states that write commands from Sinclair’s host provide logical addresses. As one example, Dr. Litz quotes Sinclair to the effect that “When the host writes data to the memory system, the controller converts the *logical addresses assigned by the host* to physical addresses within a metablock,” Ex. 1003 ¶ 199 (quoting Ex. 1007 [Sinclair] 13:9-15) (emphasis added). As another example, he states that “Sinclair discloses a controller which may receive a write request from the host, *where that host includes a logical address*,” Ex. 1003 ¶ 199 (emphasis added). The same passage acknowledges that Sinclair’s memory controller receives these write commands.

Ex. 2020 [Jones-Decl.] ¶¶ 116-117.

The problem with Petitioner’s read, of course, is that the thing that “designates a specific one of the subdivisions” in Petitioner’s identification is a metablock address—which is generated at the Sinclair’s controller, and does not accompany the request. As Dr. Jones further explains:

Dr. Litz also expressly recognizes the role of Sinclair’s memory controller in *converting* logical addresses (such as those from write commands received from Sinclair’s host) into metablock addresses. In

a passage that I have already quoted once (but now with different added emphasis), Dr. Litz quotes Sinclair to the effect that “When the host writes data to the memory system, *the controller* converts the logical addresses assigned by the host to physical addresses within a metablock,” Ex. 1003 ¶ 199 (quoting Ex. 1007 [Sinclair] 13:9-15) (emphasis added). Dr. Litz does not explain how what Petitioner identifies as a “write request from a host”—a command containing a logical address that the controller converts to a metablock address—could be said to be “accompanied by” something that the controller (the recipient of the alleged request) has to generate upon receiving the request.

To be clear, Dr. Litz does not use the claim language “accompanied by” in this paragraph of his declaration, Ex. 1003 ¶ 199, and it is not clear to me if Dr. Litz even takes this position. But to any extent that he does, I disagree. A POSITA would not understand a write command directed by Sinclair’s host to Sinclair’s controller to be “accompanied by” a metablock address that the controller generates upon receiving the command. Here, the understanding of the POSITA is consistent with a plain English understanding of the term “accompanied by,” and nothing in the knowledge of POSITA would displace the plain meaning of this term in this context.

Ex. 2020 [Jones-Decl.] ¶¶ 119-120. As Dr. Jones further explains,

There are reasons beyond the plain meaning of “accompanied by,” why a POSITA would not understand a write request from a host

to be “accompanied by” the address of a metablock that is written to as a result of the request.

As one example, even though Sinclair’s logical addresses may ultimately resolve to physical addresses, a POSITA would understand that this correspondence may not always be known in advance of a write command. In both Sinclair and other systems, a POSITA would recognize that a host may issue a write request over a large logical address range at a time when only a small amount of free physical space is available. In such situations, a POSITA would recognize that garbage collection or related maintenance operations may need to take place before the write command can be completed. In such situations, the physical locations that ultimately impacted by the write command may only be determined *after* the command has issued, and indeed, after any requisite maintenance operations are complete.

Ex. 2020 [Jones-Dec1.] ¶¶ 121-122.

Sinclair expressly refers to type of situation described by Dr. Jones when it says that “[a]n example of where execution of a host command can be delayed is where there are not enough pre-erased metablocks in the erased block pool to store all the data that the host wants to write into the memory and data consolidation or garbage collection is needed first to clear one or more metablocks of valid data, which can then be erased.” Ex. 1007 [Sinclair] 14:27-32. In this situation, a POSITA would recognize that the host providing the write command does not know which metablocks will be implicated by the logical addresses that it wants to write to. Ex.

2020 [Jones-Decl.] ¶ 123. A POSITA would further understand that such metablocks would be determined by Sinclair’s controller during or after any necessary data consolidation or garbage collection. *Id.*

For at least these reasons, Petitioner does not show that Sinclair teaches the claimed “write request from a host ... the write request accompanied by an address that designates a specific one of the subdivisions.”

2. Sinclair Does Not Render Obvious The Claimed ““write request from a host ... the write request accompanied by an address that designates a specific one of the subdivisions” In Petitioner’s Combinations.

In apparent acknowledgment of this gap between the claim and Sinclair, Petitioner states, in the alternative:

To the extent Sinclair does not teach this limitation, a POSITA would be motivated to incorporate the teaching from Son regarding write requests associated with a particular superblock, as discussed *supra* for Ground I, Claim 1[c].

Pet., 48. (“Ground I” is, of course, an Ellis ground.) This does not close the gap between Petitioner’s combination and the claim for at least two reasons.

First, Ellis and Sinclair are different systems, and however Petitioner might have argued that Son teaches a modification of *Ellis*, one cannot show that Son teaches a modification to *Sinclair* simply by citing that argument and saying that the argument would be the same for Sinclair. *See In re Magnum Oil Tools Int’l, Ltd.*,

829 F.3d at 1380 (“To satisfy its burden of proving obviousness, a petitioner cannot employ mere conclusory statements”). The fact that Petitioner’s expert essentially repeats Petitioner’s assertions, *see* Ex. 1003 [Litz-Decl.] ¶ 201-202, does not alter this conclusion. *See, e.g.*, 37 C.F.R. § 42.65(a); *see also Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9, 15-17 (Aug. 24, 2022) (precedential) (denying institution when expert declaration “adds little to the conclusory assertion for which it is offered to support”).

Second, as explained elsewhere, *see* Section II, Petitioner does not adequately articulate how a POSITA would combine *Ellis* with Son, let alone be motivated to do so with a reasonable expectation of success. Thus, even if Petitioner’s Ellis-Son arguments were not irrelevant to Sinclair, they fail for the reason that Petitioner did not show how a POSITA would incorporate Son’s teachings into Ellis.

B. Petitioner Does Not Show That Its Sinclair Combinations Render Obvious “logic to send the host the information indicating extent of page utilization of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.”

Claim 1 additionally requires “a memory controller comprising ... logic to send the host the *information indicating extent of page utilization* of the specific one of the subdivisions and *an address corresponding to the specific one of the*

subdivisions.”⁸ Petitioner identifies Sinclair’s controller as the claimed “memory controller,” *see, e.g.*, Pet., 46, a host interacting with Sinclair’s memory controller as the claimed “host,” *see, e.g., id.*, Sinclair’s “amount of valid data” and/or “number of pages of unprogrammed capacity” (or “additional variants of superblock metadata disclosed in Son”) for the claimed “information indicating extent of page utilization,” *see, e.g., id.*, 47, one of Sinclair’s “metablocks” for the claimed “specific one of the subdivisions,” *id.*, and a “metablock address” in Sinclair for the claimed “address corresponding to the specific one of the subdivisions.” *Id.*, 48.

Having made each of these identifications between Sinclair and individual claim elements, Petitioner does not assert that Sinclair’s memory controller comprises “logic to send the host the information *indicating extent of page utilization* of the specific one of the subdivisions and an address corresponding to the specific one of the subdivisions.” The evident problem is that Sinclair does not disclose its memory controller sending its host what Petitioner identifies as the claimed “information” for any metablock in Sinclair, let alone alongside what Petitioner identifies as the claimed “address corresponding to [a] specific one of the subdivisions.”

⁸ Again, all of Petitioner’s Sinclair grounds rely upon Petitioner’s arguments involving Sinclair, Son, and Yu against independent claim 1. *See* Section IV *supra*.

To supply a mechanism by which a modified Sinclair controller could send the items it has identified to the host, Petitioner turns to Yu. *See generally* Pet., 48-49. Petitioner states that Yu discloses “S.M.A.R.T. commands,” which are a mechanism by which the host *in Yu* can query a memory controller *in Yu* for “‘important data from the [controller]’ including ‘health information,’ ‘wear,’ ‘wear-leveling count,’ ‘bad block, and other flash memory management’ responsibilities.” Pet., 49 (quoting Ex. 1008 [Yu] [0064] & claim 6). Petitioner argues that Yu “teaches the use of a ‘page status table,’ as well as other metrics representing page utilization,” and by analogy with what it has already identified in Sinclair, argues that a POSITA would understand that Sinclair’s host and controller could use what Petitioner terms “Yu’s S.M.A.R.T. functionality” for its host to query the controller for, and for the controller to then send, what the claim recites. *See, e.g.*, Pet. 49-50.

Of course, if one were simply seeking to use claim 1 as a roadmap to select disclosures from Petitioner’s cited art, and one had already chosen to identify Sinclair’s metablocks with “subdivisions,” a certain data in Sinclair as “information indicating extent of page utilization,” and a metablock address in Sinclair as an “address corresponding to the specific one of the subdivision,” then if a hypothetical response to one of Yu’s S.M.A.R.T. commands is going to provide the claimed “logic to send the host information,” it would certainly help if Yu’s S.M.A.R.T. command were to request, and the response to the command to include, the information that

claim would point to under these identifications (*i.e.*, Sinclair’s “information indicating extent of page utilization” for a specific metablock).

Curiously, this *is all that Petitioner argues about why S.M.A.R.T. commands in Yu* teach the controller sending the claimed “information indicating extent of page utilization of the specific one of the subdivisions” in the combination. Pet., 49-50 (“[a] POSITA would understand, in order to implement Yu’s S.M.A.R.T. functionality in the combined system of Sinclair and Son, that the S.M.A.R.T. information sent to the host would include the combined system’s metablock-level metadata such as page utilization”) (citing Ex. 1003 [Litz-Decl.] ¶ 207). The only support identified for this conclusory assertion is Petitioner’s expert, who cites disclosures in Yu for propositions such as “Yu also teaches the maintenance of page utilization data” and “[a] POSA would understand that the page status table [as in Yu] represents page utilization.” Ex. 1003 [Litz-Decl.] ¶ 207. These generic statements do not support Petitioner’s point; they do not, for example, even show where or how Yu teaches returning its “page status table” or information from its “page status table” in response to a “S.M.A.R.T. command” *in Yu*.

Similarly, as claim 1 further requires sending the host “the information indicating extent of page utilization of the specific one of the subdivisions *and an address corresponding to the specific one of the subdivisions*,” then if Petitioner’s mappings of Sinclair are taken for granted, the use of claim 1 as a roadmap for Yu

would suggest that the hypothetical response to Yu’s S.M.A.R.T. command also include Sinclair’s metablock address—so that the alleged host is sent the claimed “address corresponding to the specific one of the subdivisions.” And again, this is *all that Petitioner argues* about why Yu’s “S.M.A.R.T. commands” teach this aspect of the combination:

Also, to apply the teaching of Yu to the combined system, a POSA would understand that the information sent by the controller to the host in connection with the S.M.A.R.T. command *must include* a superblock or metablock address, as described for Ground I, Claim 1[d]-[f].

Pet., 50 (citing Ex. 1003 [Litz-Decl.] ¶ 207). No support is offered for this assertion but a citation to Petitioner’s expert, who again does not support the relevant point. *See* Ex. 1003 [Litz-Decl.] ¶ 207. The closest the expert comes to supporting Petitioner’s point, in the paragraph immediately preceding the one cited by Petitioner, is with a conclusory statement that “a POSITA would rely on metablock-level metadata where that includes an address corresponding to the specific one of the subdivisions *so that the information is useful to the host.*” Ex. 1003 [Litz-Decl.] ¶ 206. But a generically stated desire to improve something is generally insufficient to show that a POSITA would have combined features of different references in a particular way. *See, e.g., ActiveVideo Networks, Inc. v. Verizon Commc’ns, Inc.*, 694 F.3d 1312, 1328 (Fed. Cir. 2012). And here, of course, there is no distance

whatsoever between what Petitioner’s expert argues would be “useful to the host” and exactly what the claim would require of Petitioner’s mapping. *See also, e.g., Samsung Elecs. Co., LTD., v. Red Rock Analytics, LLC*, IPR2018-00556, Paper 18, 21 (Aug. 20, 2018) (denying institution) (finding testimony that a proposed modification would provide “increased utility” to be “akin to stating in a conclusory fashion that the combination ‘would have been obvious’”).

Petitioner’s subsequent citation to “Ground I, Claim 1[d]-1[f]” in support of its argument about “the teaching of Yu” does not cure these defects. Petitioner’s Ground I does not involve Yu or Sinclair at all, and does not address why a response to a Yu S.M.A.R.T. command in Petitioner’s Ground IV combination would include a “superblock or metablock address” in a modified version of Sinclair’s system. Sinclair itself does not teach its controller sending this kind of information to its host, or Petitioner would have had no need for Yu. And Yu itself does not teach the use of Son’s “superblocks” or Sinclair’s “metablocks,” let alone the use of superblock or metablock addresses in response to “S.M.A.R.T. commands,” let alone specifically reporting page utilization information at the level of “superblocks” or “metablocks” in response to “S.M.A.R.T. commands.” And Petitioner does not even *argue*, let alone *show*, that Yu teaches these things. Petitioner only says that “to apply the teaching of Yu to the combined system, a POSITA would understand” that the relevant claim limitation would be satisfied. Pet., 50. Obviousness cannot be

shown by conclusory statements, *In re Magnum Oil Tools*, 829 F.3d at 1380, and Petitioner's Sinclair grounds therefore fail to show that this limitation is taught.

C. Petitioner Does Not Show That Its Sinclair Combinations Teach Or Render Obvious “an erasure request from the host . . . the erasure request accompanied by the address corresponding to the specific one of the subdivisions.”

The claims further require “an erasure request from the host ... the erasure request accompanied by the address corresponding to the specific one of the subdivisions.”⁹ Petitioner contends that “Sinclair teaches that the host issues, and the controller has logic to receive, ‘delete’ or erasure commands.” Pet., 50 (citing Ex. 1007 [Sinclair] 21:50-57, 27:61-28:12). But Sinclair does not teach that any such erasure commands are accompanied by what Petitioner alleges to be an “address corresponding to the specific one of the subdivisions,” which the challenged claims require. Petitioner identifies the claimed “address” with an “address designating a single metablock” in Sinclair. Pet., 50. But Petitioner does not show why any host-issued erasure request in Sinclair would include such a thing, and a straightforward analysis of the disclosures in Sinclair that Petitioner is citing reveals Petitioner's failure to prove that this limitation is satisfied.

⁹ Again, all of Petitioner's Sinclair grounds rely upon Petitioner's arguments involving Sinclair, Son, and Yu against independent claim 1. *See* Section IV *supra*.

Petitioner's first attempt to support its argument is via citation to a discussion in Sinclair of inserting new data in a file, where the updated file is stored as a new file and "[t]he old file is then deleted by the host, and the system may respond by reclaiming the logical address space assigned to the old file is stored, the data of which are now obsolete." *See* Pet. 50 (citing Ex. 1007 [Sinclair] 21:50-57). This does not disclose Sinclair's host making any use of metablock addresses relating to where the file was (or will be) stored, let alone disclose Sinclair's host issuing an erasure request "accompanied by" an specifying an address corresponding to a metablock. As Dr. Jones explains:

When a file is deleted within a file system, a host typically updates structures such as file allocation tables (FATs) or other records to indicate that a file is no longer in use and that its logical addresses are free for reassignment. A POSITA would understand that such operations typically occur at the level of the file system or a logical address space that is visible to the host, but do not generally entail any action at the level of physical management of flash memory. (In fact, the physical data of a deleted file in a system like Sinclair would typically remain resident in the flash medium until the controller, at some later time, decides that erasing the space is beneficial. *See, e.g.,* Ex. 1007 [Sinclair] 12:20-30.)

Sinclair's statement that when an "old file is then deleted by the host, [] the system may respond by reclaiming the logical address space assigned to the old file" is consistent with this general understanding.

A POSITA would not understand “reclaiming the logical address space assigned to the old file” to involve the erasure by Sinclair’s controller of the physical blocks that formerly corresponded to the logical address space occupied by the old file.

Ex. 2020 [Jones-Decl.] ¶¶ 139-140.

Petitioner’s second citation is to Ex. 1007 [Sinclair] 27:61-28:12, which is a general enumeration of circumstances in Sinclair’s system in which “data may become obsolete,” where such circumstances include “a file is deleted by a host” or “data within a file is deleted by a host.” *Id.*, 28:1-2. Again, Petitioner does not explain how Sinclair’s host deleting a *file* connects to Sinclair’s host allegedly sending erasure requests “accompanied by the address corresponding to the specific one of the subdivisions” (which Petitioner identifies with Sinclair’s “metablocks”).

Moreover, in Sinclair’s system, the translation from logical addresses to physical addresses is generally performed by the controller with a logical address as an input, and there would be no reason for Sinclair’s host to send metablock addresses to a controller capable of generating that information. Ex. 2020 [Jones-Decl.] ¶ 143. Indeed, Sinclair expressly explains that “[t]he controller of the memory system will typically change the physical block into which any particular logical block is mapped but *this is not known to the host* and does not affect its assignment of addresses of file data to the logical blocks.” Ex. 1007 [Sinclair] 18:43-

47.

Petitioner never attempts to explain why or how something that Sinclair discloses as “not known to the host” would somehow still be sent by Sinclair’s host. Petitioner simply ignores Sinclair’s statement that “the physical block into which any particular block is mapped . . . is not known to the host,” because Petitioner’s reading of claim 1 requires a “host” to specify a “metablock.”

Petitioner attempts to patch this gap between the claim and Sinclair by incorporating its arguments about its Ellis-Son combination by reference. Pet., 51. This argument fails for reasons already articulated: Petitioner cannot establish a property of a combination of Sinclair and Son purely by reference to a combination of Son with some other reference. *See, e.g.*, Sections III.C.3, III.A.2, IV.B *supra*.

V. CONCLUSION

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

Respectfully submitted,

Dated: October 20, 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 13686 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,

LOWENSTEIN & WEATHERWAX LLP

Dated: October 20, 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

EXHIBITS 2020-2030

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

James M. Glass
John T. McKee
Nima Hefazi
Gregory M. Miraglia

jimglass@quinnemanuel.com
johnmckee@quinnemanuel.com
nimahefazi@quinnemanuel.com
gregmiraglia@quinnemanuel.com

Respectfully submitted,

Dated: October 20, 2025

By: /s/ Michael Cosley
Michael Cosley