

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

KINGSTON TECHNOLOGY COMPANY, INC., KINGSTON TECHNOLOGY
CORPORATION, AND KINGSTON DIGITAL, INC.,
Petitioners,

v.

VERVAIN, LLC,
Patent Owner.

IPR2025-00614
U.S. Patent No. 8,891,298

**PATENT OWNER'S PRELIMINARY RESPONSE ON THE MERITS AND
OTHER NON-DISCRETIONARY CONSIDERATIONS**

TABLE OF CONTENTS

	Page
EXHIBIT LIST	vi
I. INTRODUCTION	1
II. OVERVIEW OF THE CHALLENGED PATENT AND THE CHALLENGED CLAIMS	2
A. SLC and MLC Flash	3
B. Address Table.....	5
C. Data Integrity Tests	6
D. Hot and Cold Data.....	6
E. Claim 1	6
III. KINGSTON’S MOTION FOR JOINDER.....	9
IV. PERSON OF ORDINARY SKILL IN THE ART	9
V. CLAIM CONSTRUCTION	10
A. “Blocks”	11
B. Other Terms.....	17
VI. OVERVIEW OF THE CITED PRIOR ART	18
A. Gavens (Ex. 1045).....	18
B. Moshayedi (Ex. 1043).....	26
C. Sutardja (Ex. 1042).....	28
VII. REASONS FOR DENYING INSTITUTION.....	30
A. Ground 1: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Gavens.....	30

1.	[1b]: “SLC non-volatile memory module comprising a plurality of <i>individually addressable blocks</i> ”	31
2.	[1c]: “a controller coupled to” an “MLC non-volatile memory module” and an “SLC non-volatile memory module”	31
3.	[1f]: “c) determine which of the blocks of the plurality of blocks...are accessed most frequently by <u><i>maintaining a count</i></u> of the number of times each one of the blocks is accessed”	36
4.	[1f]: “ <i>controller</i> adapted to” ([1c]) ... “maintain[] a count of the number of times each of the blocks is accessed”	39
5.	[1g]: “allocate those blocks that receive the most frequent writes by <u><i>transferring the respective contents of those blocks</i></u> to the at least one SLC non-volatile memory module”	41
B.	Ground 2: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Moshayedi.....	44
1.	[1b]: “at least one SLC non-volatile memory module comprising a plurality of <i>individually erasable blocks</i> ”	44
2.	[1d]: “a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a list of logical address ranges accessible by a computer system, the list of logical address ranges having a <i>minimum quanta of addresses</i> ,...”	45
3.	[1e]: “b) determine if a range of addresses listed by an entry and mapped to a similar range of physical addresses within the at least one <u><i>MLC non-volatile memory module</i></u> , <i>fails a data integrity test</i> , and, in the event of such a failure, the controller <u><i>remaps the entry to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module</i></u> ”	46

4.	[1g]: “d) allocate those blocks that receive the most frequent writes by <i>transferring the respective contents of those blocks</i> to the at least one SLC non-volatile memory module”	48
C.	Ground 3: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Sutardja	52
1.	[1b]: “at least one SLC non-volatile memory module comprising a plurality of <i>individually erasable blocks</i> ”	53
2.	[1d]: “a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a list of logical address ranges accessible by a computer system, the list of logical address ranges having a <i>minimum quanta of addresses,...</i> ”	53
3.	[1e]: “b) determine if <i>a range of addresses listed by an entry</i> and mapped to a similar range of physical addresses within the at least one MLC non-volatile memory module, fails a data integrity test, and, in the event of such a failure, the controller <i>remaps the entry</i> to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module”	54
4.	[1f]: “c) determine which of the blocks of the plurality of the blocks in the MLC and SLC non-volatile memory modules are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed”	57
5.	[1g]: “d) allocate those blocks that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module”	58
VIII.	CONCLUSION.....	63

TABLE OF AUTHORITIES

	Page(s)
CASES	
<i>A.C. Dispensing Equip. Inc. v. Prince Castle LLC</i> , IPR2014-00511, Paper 6 (P.T.A.B. Sept. 10, 2014).....	49, 58, 59
<i>Abiomed, Inc. v. Maquet Cardiovascular, LLC</i> , IPR2017-01204, 2017 WL 4804523 (P.T.A.B. Oct. 23, 2017).....	38, 52, 61
<i>Acon Inc. v. Amo Dev., LLC</i> , IPR2021-00858, 2022 WL 17585777 (P.T.A.B. Dec. 12, 2022).....	45, 53, 54
<i>Application of Ratti</i> , 270 F.2d 810 (C.C.P.A. 1959)	36
<i>Captioncall, LLC v. Ultratec, Inc.</i> , No. IPR2020-01215, 2021 WL 278381 (P.T.A.B. Jan. 27, 2021)	37
<i>DeSilva v. DiLeonardi</i> , 181 F.3d 865 (Fed. Cir. 1999)	33, 45
<i>GN Resound A/S v. Oticon A/S</i> , IPR2015-00103, Paper 13 (P.T.A.B. Jun. 18, 2015)	31, 49, 58, 59
<i>In re Kahn</i> , 441 F.3d 977 (Fed. Cir. 2006)	33
<i>In re Stepan Co.</i> , 868 F.3d 1342 (Fed. Cir. 2017)	34, 43, 48, 56
<i>Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	43
<i>Masimo Corp. v. Apple Inc.</i> , IPR2023-00745, Paper 44 (P.T.A.B Oct. 15, 2024).....	38
<i>Nearmap US, Inc. v. Eagle View Technologies, Inc.</i> , IPR2024-00716, Paper 9 (P.T.A.B. Oct. 9, 2024).....	31, 44, 53
<i>Netflix, Inc. v. DivX, LLC</i> , 84 F.4th 1371 (Fed. Cir. 2023)	43

<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Circ. 2005) (<i>en banc</i>)	10
<i>Phison Electronics Corporation v. Vervain, LLC</i> , IPR2025-00212, Paper 10 (P.T.A.B. May 28, 2025)	9
<i>Samsung Elecs. Co., Ltd. v. Mojo Mobility Inc.</i> , IPR2023-01099, 2024 WL 457731 (P.T.A.B. Feb. 6, 2024)	36
<i>Toyota Motor Corp. v. Cellport Systems, Inc.</i> , IPR2015-00633, 2015 WL 4934778 (P.T.A.B. Aug. 14, 2015).....	10
<i>Vervain, LLC v. Phison Electronics Corp.</i> , 1:24-cv-00259 (W.D. Tex.)	10, 12
<i>Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.</i> , 200 F.3d 795 (Fed. Cir. 1999)	17

STATUTES

35 U.S.C.	
§ 312(a)(3)	31, 33, 45, 46
§ 313.....	1
§ 314(a).....	1

OTHER AUTHORITIES

37 C.F.R.	
§ 42.6(a)(3)	45
§ 42.22(a)(2)	33
§ 42.107.....	1
§ 100(b).....	10

EXHIBIT LIST

Exhibit No.	Description	Previously Submitted
2001	Declaration of Dr. Sunil Khatri from <i>Phison Electronics Corporation v. Vervain, LLC</i> , IPR2025-00212, Ex. 2001 (P.T.A.B. Mar. 26, 2025) (expert report filed with Patent Owner’s Preliminary Response)	X
2002	<i>Vervain, LLC v. Western Digital Corp. et al.</i> , No. 6:21-cv-488-ADA, Dkt. 41 (W.D. Tex. Jan. 24, 2022) (Claim Construction Order)	X
2003	<i>Vervain, LLC v. Kingston Technology Company, Inc., et al.</i> , No. 1:24-cv-254-ADA, Dkt. 19 (W.D. Tex. Jul. 15, 2024) (agreed scheduling order)	X
2004	<i>Vervain, LLC v. Kingston Technology Company, Inc., et al.</i> , No. 1:24-cv-254-ADA, Dkt. 56 (W.D. Tex. Jan. 28, 2025) (order denying motion to transfer venue)	X
2005	<i>Vervain, LLC v. Phison Electronics Corporation</i> , No. 1:24-cv-259-ADA, Dkt. 16 (W.D. Tex. Jul. 15, 2024) (agreed scheduling order)	X
2006	“Judge Albright Patent FAQ,” https://www.txwd.uscourts.gov/for-attorneys/judge-albright-courtroom-faq/ . Accessed December 18, 2024.	X
2007	Judge Albright’s Standing Order Governing Proceedings (OGP) 4.4—Patent Cases, Filed January 23, 2024.	X
2008	Preliminary Claim Constructions for <i>Vervain, LLC v. Phison Electronics Corporation</i> , No. 1:24-cv-259-ADA and <i>Vervain, LLC v. Kingston Technology Company, et al.</i> , No. 1:24-cv-254-ADA – email from Mr. Brown received on February 5, 2025.	X
2009	Markman Hearing Transcript dated February 6, 2025, for <i>Vervain, LLC v. Phison Electronics Corporation</i> , No. 1:24-cv-259-ADA and	X

Exhibit No.	Description	Previously Submitted
	<i>Vervain, LLC v. Kingston Technology Company, et al.</i> , No. 1:24-cv-254-ADA.	
2010	<i>Vervain, LLC v. Western Digital Corp. et al.</i> , No. 6:21-cv-00488-ADA, Dkt. 180 (W.D. Tex. Jul. 26, 2023) (Redacted Copy of Order on the Pending Motions and Motions <i>In Limine</i>).	X
2011	<i>Phison Electronics Corp. v. Vervain, LLC</i> , PGR2024-00047, Paper 9 (P.T.A.B. Mar. 17, 2024) (Decision Denying Institution of Post-Grant Review of U.S. Patent No. 11,830,546)	X
2012	<i>Phison Electronics Corp. v. Vervain, LLC</i> , PGR2024-00048, Paper 9 (P.T.A.B. April 28, 2025) (Decision Denying Institution of Post-Grant Review of U.S. Patent No. 11,854,612)	X
2013	Docket Sheet for <i>Vervain, LLC v. Kingston Technology Company, Inc., et al.</i> , No. 1:24-cv-254-ADA (W.D. Tex.). Accessed May 19, 2025.	X
2014	Excerpts from Kingston’s Final Invalidity Contentions served on February 28, 2025 in <i>Vervain, LLC v. Kingston Technology Company, Inc., et al.</i> , No. 1:24-cv-254-ADA (W.D. Tex.)	X
2015	“Interim Processes for PTAB Workload Management Memorandum,” issued March 26, 2025, https://www.uspto.gov/sites/default/files/documents/InterimProcesses-PTABWorkloadMgmt-20250326.pdf . Accessed May 13, 2025.	X
2016	Docket Sheet for <i>Vervain, LLC v. Phison Electronics Corporation</i> , No. 1:24-cv-259-ADA (W.D. Tex.). Accessed May 19, 2025.	X
2017	<i>Vervain, LLC v. Western Digital Corp. et al.</i> , No. 6:21-cv-00488-ADA, Dkt. No. 183, Pretrial Hearing Redacted Transcript (W.D. Tex. Aug. 7, 2023).	
2018	Chen et al., <i>Ultra MLC Technology Introduction</i> , Advantech Technical White Paper (Oct. 5, 2012) (“Chen”)	

Exhibit No.	Description	Previously Submitted
2019	Excerpts from Micheloni et al., <i>Inside NAND Flash Memories</i> (1 st ed. 2010) (“Micheloni”)	
2020	<i>Vervain v. Western Digital Corp. et al.</i> , No. 6:21-cv-00488-ADA, Dkt. 180 (W.D. Tex. Jul. 26, 2023) (Redacted Copy of Order on the Pending Motions and Motions <i>In Limine</i>).	

I. INTRODUCTION

Vervain, LLC (“PO” or “Vervain”) submits this Preliminary Response in accordance with 35 U.S.C. § 313 and 37 C.F.R. § 42.107, responding to the Petition for *Inter Partes* Review (IPR) (Paper 1) of U.S. Patent No. 8,891,298 (“298 Patent” or “the Challenged Patent”) filed by Kingston Technology Company, Inc., Kingston Technology Corporation, and Kingston Digital, Inc. (“Kingston” or “Petitioners”). The Petition includes three grounds of unpatentability challenging claims 1-11 (“Challenged Claims”). Each ground relies on single-reference obviousness, based on U.S. Patent No. 8,634,240 (“Gavens”) (Ex. 1045), U.S. Patent Appl. Pub. No. 2009/0327591 (“Moshayedi”) (Ex. 1043), and U.S. Patent Appl. Pub. No. 2008/0140918 (“Sutardja”) (Ex. 1042), respectively.

The Board should deny institution because Petitioner’s grounds are meritless. Each ground is cursory and inadequately explained. Petitioner omits certain limitations from its analysis entirely, and where limitations are addressed, the Petition fails to reconcile critical differences between the Challenged Claims and the prior art. Petitioner has not shown a reasonable likelihood of prevailing on any ground. *See* 35 U.S.C. § 314(a).

Accordingly, PO requests that the Board deny institution of the Petition.

Further, Petitioner moved to join this proceeding (IPR2025-00614) with IPR2025-00212. *See* Paper 3. Because the Board already denied institution of IPR2025-00212, Petitioner’s motion for joinder is now moot.

II. OVERVIEW OF THE CHALLENGED PATENT AND THE CHALLENGED CLAIMS

The Challenged Patent, entitled “Lifetime Mixed Level Non-Volatile Memory System” was filed on April 25, 2012, and has an effective filing date of July 19, 2011. Ex. 1001. Dr. G. R. Mohan Rao is the sole named inventor of the Challenged Patent.

At a high level, the Challenged Patent describes a reliable flash memory storage system combining both single-level cell (SLC) and multi-level cell (MLC) non-volatile memories.¹ Ex. 1001, Abstract; Ex. 2001, ¶38. Prior to the Challenged Patent, Dr. Rao recognized that “MLC NAND flash SSDs are slowly replacing and/or coexisting with SLC NAND flash in newer SSD systems” because “MLC flash memory is less expensive than SLC flash memory[] on a cost per bit basis.” Ex. 1001, 3:14-15, 5:24-26. However, while “MLC NAND flash enjoys greater density than SLC NAND flash” it comes “at the cost of a decrease in access speed and lifetime (endurance).” *Id.*, 3:19-21. As a result, various hybrid systems

¹ Flash memory is a specific type of non-volatile memory, where data is stored in “blocks” of “pages.” Ex. 1001, 2:31-48; Ex. 2001, ¶27.

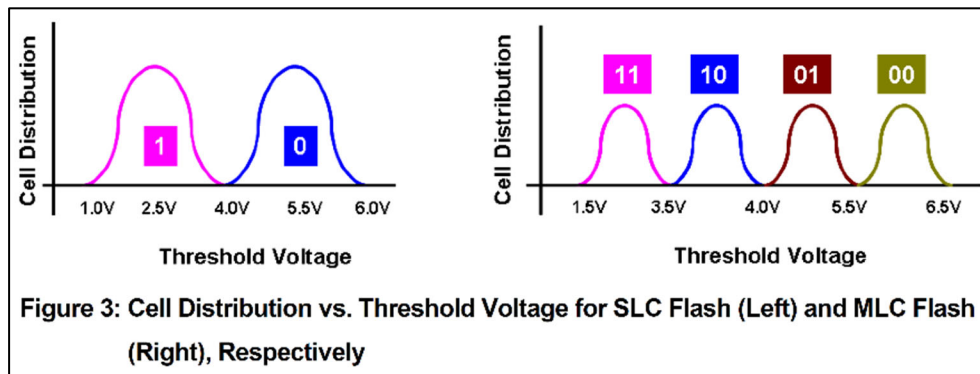
combining SLC and MLC (among others) are taught in the Challenged Patent, to combine the benefits of both types of non-volatile flash storage at a low cost. *Id.*, 3:43-45

The Challenged Patent addresses improvements and solutions for managing the writing of data optimally for improved reliability and lifetime (endurance) of such hybrid memory systems. *Id.*, 3:38-45. Specifically, the Challenged Claims are directed to techniques for efficiently using SLC and MLC flash to improve the overall performance of the memory. *Id.*, claim 1. For example, if certain data is used more frequently, then it is transferred to higher-performance SLC. *Id.* By doing so, the number of errors is reduced, and overall endurance of the memory is increased. *Id.*, 3:43-45.

A. SLC and MLC Flash

SLC memory stores 1 bit per cell, and MLC memory stores more than 1 bit per cell. Ex. 1001, 1:64-67; Ex. 2001, ¶¶32-34. As noted above, there are advantages and disadvantages to SLC and MLC flash. In general, SLC is faster and less prone to errors, but requires more space and power to store a given amount of data. Ex. 1001, 1:38-43. The opposite is true of MLC. MLC flash is slower and more prone to errors, but stores data more densely with less power consumption. *Id.*, 3:19-21.

SLC and MLC flash memories both use floating gate transistors. *Id.*, 3:29. They store a charge in the floating gate of each transistor (cell), which changes the threshold voltage of the transistor. Ex. 2001, ¶32. The memory uses the threshold voltage to determine what bit, or bits, were stored in the transistor. The MLC cell in the figure below illustrates threshold voltages for a 2-bit MLC cell.



Ex. 2018, 3.

The primary difference between SLC and MLC is what data each threshold voltage represents. Ex. 2001, ¶33. With SLC flash, the transistor stores only a 1 or 0, so a wide range of threshold voltages can be allotted to a single bit. Ex. 1001, 3:15-17. This allows for faster and more reliable memory access. On the other hand, MLC flash must be slowly and carefully programmed to a narrower, more precise range of threshold voltages, with each threshold voltage range representing a specific pair of bits (*see* figure above, which shows four pairs of bits—11, 10, 01, and 00—corresponding to smaller ranges of threshold voltages compared to the SLC). Ex. 1001, 3:17-19.

B. Address Table

To provide wear leveling, garbage collection, and bad block management, a translation layer is used to map logical addresses to actual physical memory locations. Ex. 2019, 40-42; Ex. 1001, 2:49-3:13; *see also* Ex. 2001, ¶35. As part of this translation layer, “tables are widely used in order to map sectors and pages from logical to physical.” Ex. 2019, 40-42; Ex. 1001, 2:64-3:1. These tables map logical blocks to physical blocks. *Id.* Using a “block” or similar granularity is important, since flash memory is arranged so that when erasing and rewriting data, a whole block is “erased together.” Ex. 2019, 22; Ex. 1001, 2:38-48. Dr. Rao explained that “[t]he address ranges within the translation table will assume some minimum quantum, such as, for example, one block....” Ex. 1001, 5:27-31, Figs. 3a-3b. Dr. Rao further explained that memory is written and mapped on the granularity of a “quantum.” *Id.*

During operation of the flash memory, logical addresses are frequently remapped to new physical locations. *Id.*, 2:65-3:31, 3:67-4:10, 5:20-40. Over time, a particular logical address may be mapped or associated with many different physical locations (blocks). Ex. 2001, ¶60. And multiple logical addresses may point to the same block over time, so there is not a one-to-one correspondence between the logical addresses and the blocks over time. *Id.*

C. Data Integrity Tests

As mentioned above, when data is stored in MLC memory, it is more prone to errors, and some data is more prone to errors than other data. Ex. 2001, ¶36. A data integrity test checks the integrity of the data (*i.e.*, whether errors have occurred). If the test reveals a problem such as corrupt data, the data can be remapped to SLC (which is less error-prone) or, alternatively, other MLC blocks, and the address table is modified accordingly. Ex. 1001, 2:59-3:13, 4:4-10.

D. Hot and Cold Data

One can distinguish between “hot” blocks (which receive more frequent writes), and “cold” blocks (which receive less frequent writes). *Id.*, 6:24-35. Because SLC has greater endurance, “hot” blocks can be allocated to SLC to increase the lifetime of the system. *Id.* “Cold” blocks, on the other hand, can be allocated to MLC to take advantage of its higher density storage. Ex. 2001, ¶37

E. Claim 1

Claim 1 requires a system for storing data comprising MLC and SLC non-volatile memory. The MLC and SLC comprise “erasable blocks” (highlighted red below). These are the physical locations that must be erased before data can be written to them. *See* [1a]-[1b] and [1f]-[1g] below. Meanwhile, an address map comprises a list of “logical address ranges” (highlighted purple); these logical address ranges are mapped to the physical address ranges for the blocks. *See* [1d]-[1e].

Claim 1	
[1Pre]	A system for storing data comprising:
[1a]	at least one MLC non-volatile memory module comprising a plurality of individually erasable blocks ;
[1b]	at least one SLC non-volatile memory module comprising a plurality of individually erasable blocks ; and
[1c]	a controller coupled to the at least one MLC non-volatile memory module and the at least one SLC non-volatile memory module wherein the controller is adapted to:
[1d]	a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a list of logical address ranges accessible by a computer system, the list of logical address ranges having a minimum quanta of addresses, wherein each entry in the list of logical address ranges maps to a similar range of physical addresses within either the at least one SLC non-volatile memory module or within the at least one MLC non-volatile memory module;
[1e]	b) determine if a range of addresses listed by an entry and mapped to a similar range of physical addresses within the at least one MLC non-volatile memory module, fails a data integrity test, and, in the event of such a failure, the controller remaps the entry to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module;

[1f]	c) determine which of the blocks of the plurality of the blocks in the MLC and SLC non-volatile memory modules are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed; and
[1g]	d) allocate those blocks that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module.

As can be seen above, Claim 1 uses the claim terms “blocks” and “logical address ranges” to refer to two different things. The blocks are the physical locations in the MLC and SLC where the data is stored. *See* [1a]-[1b]. Each block has a fixed “range of physical addresses.” *See* [1d]. Meanwhile, the address map contains a list of logical address ranges that are mapped to the physical address ranges. *Id.* As the claim indicates, the logical address ranges are remapped to new physical address ranges. *See* [1e]. Thus, a logical address range does not permanently point to a specific physical address range. Rather the corresponding physical address range may change over time.

Turning to [1f], the claim refers to “the blocks,” where the antecedent basis is “erasable blocks” in [1a]-[1b]. Thus, the controller is adapted to “determine which of the [erasable blocks]...are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed.”

Finally, in [1g], the controller is adapted to transfer the contents of those blocks that receive the most frequent writes to SLC memory.

III. KINGSTON’S MOTION FOR JOINDER

Phison Electronics Corporation (“Phison”) previously filed a request for *inter partes* review of the Challenged Patent in IPR2025-00212. Kingston represented that its Petition in this proceeding (IPR2025-00614) includes arguments identical to those in Phison’s petition (IPR2025-00212) and, accordingly, filed a motion for joinder. *See, e.g.*, Paper 3 (Motion for Joinder to Phison’s IPR2025-00212 to assume an “understudy” role).

The Board has since denied institution in IPR2025-00212. *See Phison Electronics Corporation v. Vervain, LLC*, IPR2025-00212, Paper 10 (P.T.A.B. May 28, 2025) (decision denying institution of *inter partes* review).

Thus, Kingston’s motion for joinder (Paper 3) is now moot.

IV. PERSON OF ORDINARY SKILL IN THE ART

For purposes of this Preliminary Response only, PO adopts Petitioner’s definition of a person of ordinary skill in the art (POSA). Petition, 17; Ex. 2001, ¶¶22-26.²

² PO submits the declaration of Dr. Sunil Khatri—an expert in the field of the Challenged Patent—previously filed with the patent owner preliminary response in

V. CLAIM CONSTRUCTION

Claim terms are afforded “their ordinary and customary meaning,” which is “the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention.” *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Circ. 2005) (*en banc*); 37 C.F.R. § 100(b). The Board, however, only construes claim terms when necessary to resolve an underlying controversy. *Toyota Motor Corp. v. Cellport Systems, Inc.*, IPR2015-00633, 2015 WL 4934778, at *8 (P.T.A.B. Aug. 14, 2015).

Petitioner proposes construction of five terms: “blocks,” “MLC non-volatile memory module,” “SLC non-volatile memory module,” “controller,” and “data integrity test.” Petition, 18-21. PO disagrees with all of Petitioner’s proposed constructions. Petitioner’s proposals mirror those already presented in the parallel district court litigation between Vervain and Phison (*Vervain, LLC v. Phison Electronics Corp.*, 1:24-cv-00259 (W.D. Tex.) (the “Parallel Litigation”)), where

IPR2025-00212. Ex. 2001, ¶¶1-26. “Kingston’s petition [in IPR2025-00614] and the petition in Phison[’s] 298 [IPR2025-00212] are substantively identical[.]” Paper 3 (Motion for Joinder to Phison’s IPR2025-00212 to assume an “understudy” role),

1.

they were fully briefed and, during a *Markman* hearing on February 6, 2025, ***already rejected by the Court***. See Exs. 2008-2009.

A. “Blocks”

PO disagrees with Petitioner’s construction of “blocks,” and contests that construction below. The Board, however, need not adopt PO’s construction of “blocks” to deny institution of this Petition. Particularly, some of Petitioner’s grounds rely on the incorrect construction of “blocks” as logical blocks. See, e.g., Petition, 30 (“it is unclear whether ‘those blocks’ are logical or physical. Both are constructions...considered here.”). As explained below, however, ***Petitioner’s grounds still fail when the incorrect “logical” blocks construction is applied***. Thus, construction of “blocks” would be unnecessary if the Board denies Petitioner’s grounds under both the “logical” and “physical” constructions for “blocks.”

Notwithstanding, as used in the Challenged Patent, “blocks” refers only to ***physical*** blocks, and should be construed as, “in a non-volatile memory, a physical group of memory cells.” Ex. 2008, 3. The district court has twice construed this term to have this meaning. See Ex. 2009, 22; Ex. 2020, 1; Ex. 2017, 71. The Board should not diverge from that construction now. Petitioner’s contention—that that the term “blocks” is “indefinite as to whether it refers to logical or physical blocks”—is wrong. Petition, 21.

This term was already fully briefed during the summary judgment phase of an earlier litigation involving the Challenged Patent, *Vervain v. Western Digital Corp. et al.* (“*WD*”), and the court adopted PO’s construction. *See* Ex. 2020, 1. This term was again fully briefed during the claim construction phase of the Parallel Litigation, where the Court once more adopted PO’s construction. *See* Ex. 2009, 22 (maintaining the court’s preliminary construction of “blocks,” included in Ex. 2008 at page 3).

Nonetheless, Petitioner still contends that it is unclear whether “blocks” refers to a “physical” block *inside* the flash memory, or a “logical” block *outside* the memory. Petition, 21. But, as explained above, the district court has already twice decided this issue in PO’s favor. Exs. 2008-2009, 2020. Far from being indefinite, the Court already agreed with PO that the term “block” refers to a physical group of memory cells in a non-volatile memory. *Id.*

Exemplary claim 298:1 recites that the MLC and SLC non-volatile memory modules comprise a “plurality of individually erasable blocks.” Thus, the claimed “blocks” must be “in the non-volatile memories.” Claim 1 is shown below:

1. A system for storing data comprising:
at least one MLC non-volatile memory module comprising a plurality of individually erasable **blocks**;
at least one SLC non-volatile memory module comprising a plurality of individually erasable **blocks**; and

a controller coupled to the at least one MLC non-volatile memory module and the at least one SLC non-volatile memory module wherein the controller is adapted to:

...

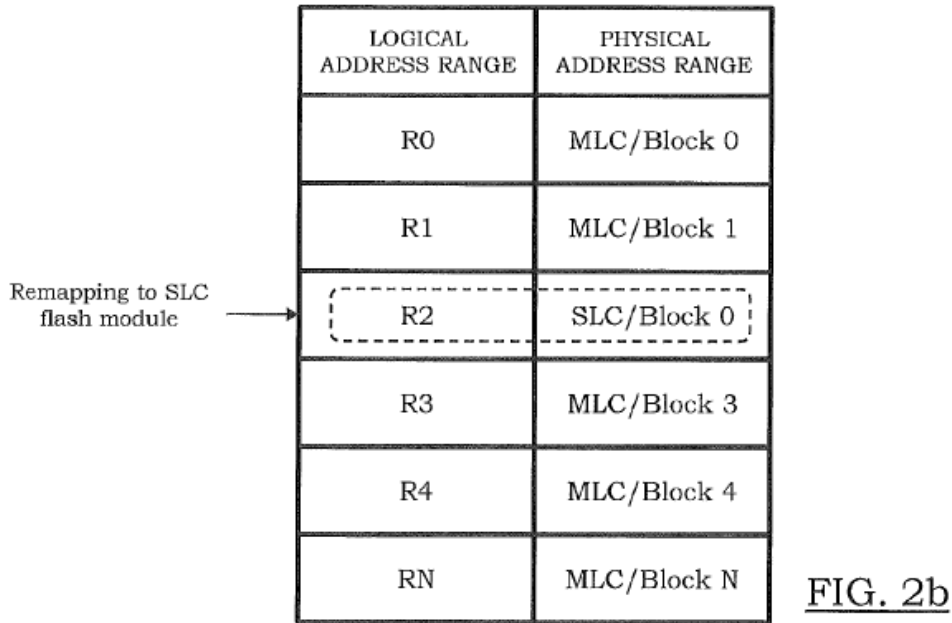
c) determine which of the **blocks** of the plurality of the blocks in the MLC and SLC non-volatile memory modules are accessed most frequently by maintaining a count of the number of times each one of the **blocks** is accessed; and

d) allocate those **blocks** that receive the most frequent writes by transferring the respective contents of those **blocks** to the at least one SLC non-volatile memory module.

Ex. 1001, 7:8-8:8 (emphasis added). As shown above, the claim uses the term “blocks” six times. In part (c), the claim specifies “*the* blocks.” Thus, the claim is referring back to *the* blocks in the SLC and MLC non-volatile memories, where *the* blocks in the SLC and MLC non-volatile memories are the antecedent basis.

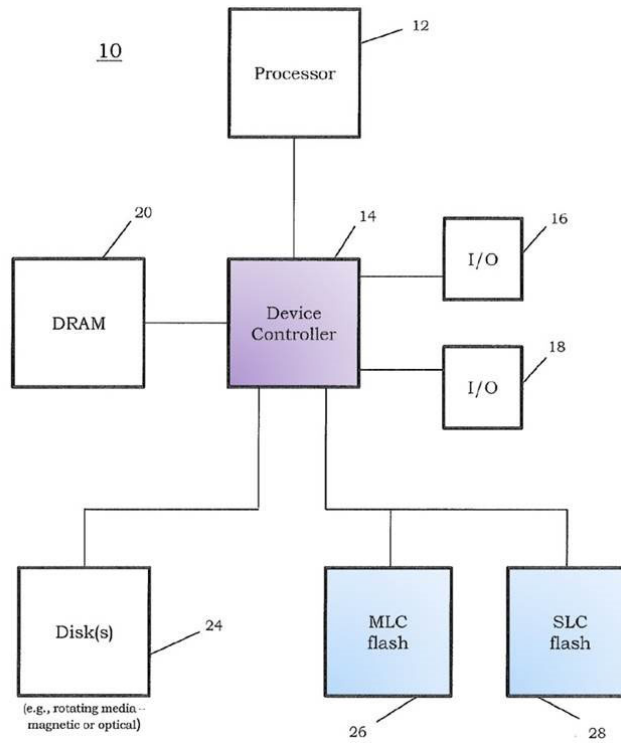
The claim also recites a controller. This controller does four things: (1) maintain an address map, (2) determine if a range of physical addresses fails a data integrity test, (3) determine which of “*the* blocks...*in the* MLC and SLC non-volatile memory modules” are accessed most frequently, and (4) transfers the contents of those blocks to the SLC non-volatile memory module.

The specification explains that the controller deals with physical blocks when it performs erase or write operations. Ex. 1001, 3:1-13 (“The controller’s wear-leveling algorithm *determines which physical block to use* each time data is programmed...”) (emphasis added). The specification describes blocks of MLC and SLC as being a physical group of memory cells, as shown below in Figure 2B:



Ex. 1001, FIG. 2b; *see also id.*, FIG. 2a, 5:44-46 (“logical address ranges are translated to **blocks on the MLC** NAND flash memory module 26”).

Petitioner assumes that “blocks” in the claims could refer to either logical blocks or physical blocks. Petition, 21. But the claims say otherwise, and make clear that the “blocks” referred to are physical blocks. For example, the claims state that the SLC and MLC non-volatile memory modules comprise “blocks.” Therefore, the blocks must necessarily be in the non-volatile memories. Otherwise, the SLC and MLC memory modules would not comprise blocks. The specification confirms that the claimed “blocks” are not “logical blocks.” *First*, Fig. 1 shows how a host processor 12 communicates with a controller 14 (shaded purple), which communicates with the flash memories 26 and 28 (shaded blue).



Ex. 1001, Fig. 1. Fig. 2b shows the flash memories in more detail. As shown below, the blocks are in the flash memories. These blocks are physical groups of memory cells; they are not logical blocks.

LOGICAL ADDRESS RANGE	PHYSICAL ADDRESS RANGE
R0	MLC/Block 0
R1	MLC/Block 1
R2	SLC/Block 0
R3	MLC/Block 3
R4	MLC/Block 4
RN	MLC/Block N

Remapping to SLC flash module →

FIG. 2b

Ex. 1001, Fig. 2b; *see also id.*, FIG. 2a. The specification explains that “logical address ranges are translated to blocks on the MLC NAND flash memory module 26.” *Id.*, 5:44-46. Thus, the blocks cannot be *logical* addresses. In other words, the logical addresses (or logical blocks) are used by the host processor 12. The controller 14 then converts the logical addresses to physical address ranges that correspond to the physical groups of memory cells.

Second, because the “blocks” are being erased by the controller, they must be physical groups of memory cells. This is because the controller can only erase physical blocks, as opposed to logical blocks. Ex. 2001, ¶¶60-61. Moreover, the controller is determining which blocks are accessed most frequently, which is part of the wear-leveling algorithm. Ex. 1001, 8:1-5. In case there is any doubt that “block” means a “physical block,” the specification makes clear that the “*controller’s* wear-leveling algorithm *determines which physical block to use* each time data is programmed....” *Id.*, 3:1-6 (emphasis added); *see also id.*, 3:28-30 (“any ‘write’ or ‘program’ to a block in NAND flash (floating gate) requires an ‘erase’ (of a block) before ‘write’”). This clear and unequivocal statement in the specification puts to rest any argument that “blocks” could be “logical blocks.”

Third, the specification’s disclosure of writing/erasing a block in “NAND flash (floating gate)” (*id.*, 3:28-29) further confirms that the MLC/SLC blocks are

physical blocks. This is because a floating gate is a physical component that is used at the physical level, not the logical level. *See, e.g.*, Ex. 2001, ¶62.

Fourth, the specification explains that “[b]locks can only be erased in their entirety, and when erased, are usually written to ‘1’ bits.” Ex. 1001, 2:43-45. This confirms that “blocks” are physical groups of memory cells, because it only makes sense to write ‘1’ bits to a physical address range in the flash memories. Additionally, this disclosure confirms that the cells in a “block” must be erased together.

Petitioner provides no substantive argument for why “blocks” should encompass “logical or physical blocks[.]” Petition, 21. Accordingly, the Board should adopt PO’s construction to clarify the scope of the claimed “blocks,” consistent with the district court’s prior rulings on this issue. *See* Exs. 2008-2009, 2020.

B. Other Terms

Petitioner proposes construction of four additional terms: “MLC non-volatile memory module,” “SLC non-volatile memory module,” “controller,” and “data integrity test.” Petition, 18-21. These proposed constructions have already been rejected by the district court. *See* Ex. 2002; Ex. 2008. PO disagrees with Petitioner’s proposed constructions, but construction of these terms is not necessary to resolve the parties’ dispute regarding whether institution should be granted. *See Vivid Techs.*,

Inc. v. Am. Sci. & Eng'g, Inc., 200 F.3d 795, 803 (Fed. Cir. 1999). Particularly, even if Petitioner's proposed constructions are applied—where many of Petitioner's grounds rely solely on their own constructions—Petitioner's grounds still fail, as explained below. Accordingly, PO does not contest Petitioner's proposed constructions for these terms for the limited purpose of this Preliminary Response.

VI. OVERVIEW OF THE CITED PRIOR ART

Petitioner asserts the following grounds:

Ground 1: Petitioner alleges that Claims 1-11 are obvious over Gavens (Ex. 1045) in view of the knowledge of a POSA (Petition, 21-38).

Ground 2: Petitioner alleges that Claims 1-11 are obvious over Moshayedi (Ex. 1043) in view of the knowledge of a POSA (Petition, 39-51).

Ground 3: Petitioner alleges that Claims 1-11 are obvious over Sutardja (Ex. 1042) in view of the knowledge of a POSA (Petition, 52-66).

A. Gavens (Ex. 1045)

Gavens discusses a problem that occurs with flash memory—as it ages, its error rate increases. Ex. 1045, 3:56-4:5; Ex. 2001, ¶¶64-80. To ensure data integrity in such situations, the memory uses a resource-intensive error correction code (ECC) to correct errors. Ex. 1045, 3:56-4:5. Gavens explains that this poses a problem because, “for most of the life time of the device [(before the device ages and the

error rate increases)], the ECC is only marginally utilized, resulting in its large overheads being wasted[.]” *Id.*, 3:67-4:2.

Gavens describes a flash memory device including **controller 102** (highlighted purple) and **memory chip 100** (highlighted teal).

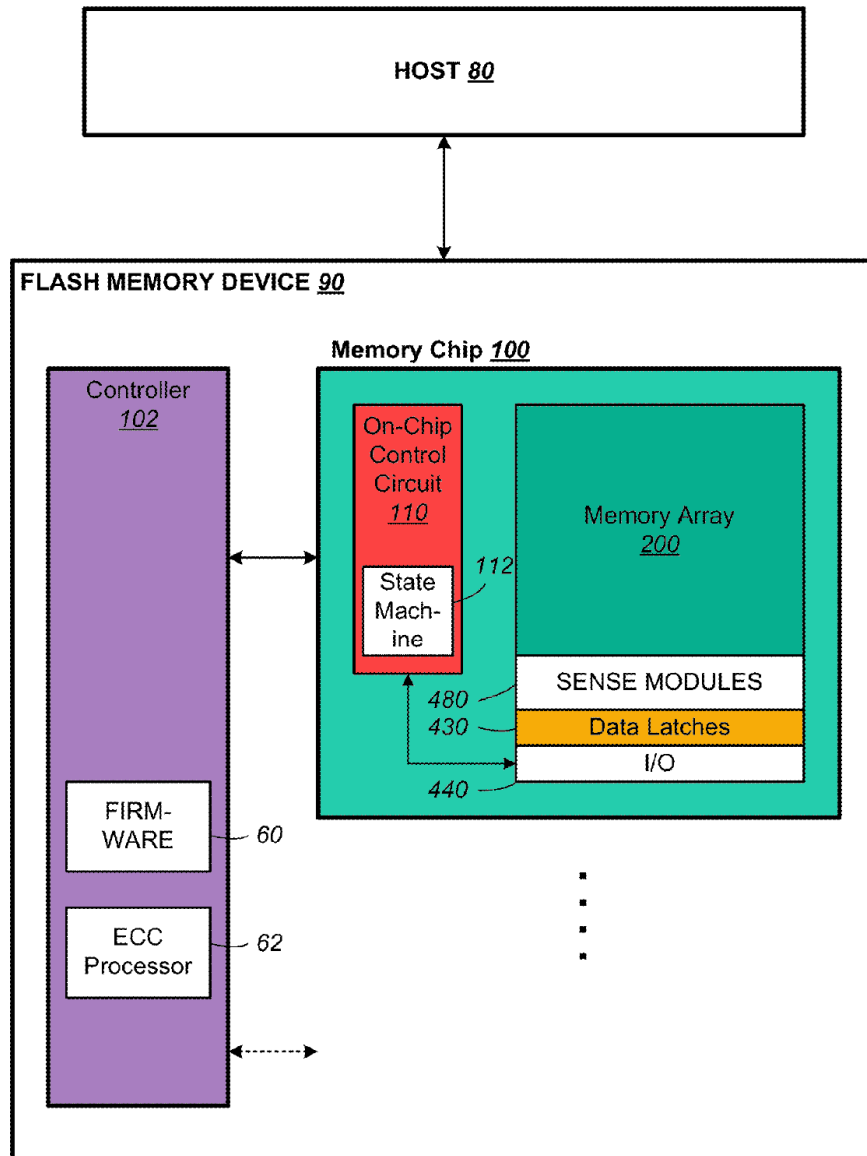


FIG. 1

Ex. 1045, Fig. 1 (annotated), 8:13-39. The **memory chip 100** also includes an **on-chip control circuit 110**. The **controller 102** “controls and manages higher level memory operations” while the **on-chip control circuit 110** controls “low-level memory operations of each chip.” *Id.*, 8:23-39. The memory chip 100 also includes **data latches 430** and **memory array 200**. *Id.*, 4:13-16, 8:13-39; Ex. 2001, ¶65.

Before writing a page of data to the memory array, “ECC is computed for the data page” by the ECC Processor 62, which is part of the **controller 102**. Ex. 1045, 12:51-63. The page of write-data then consists of two portions: the “user data” and the computed ECC. *Id.*

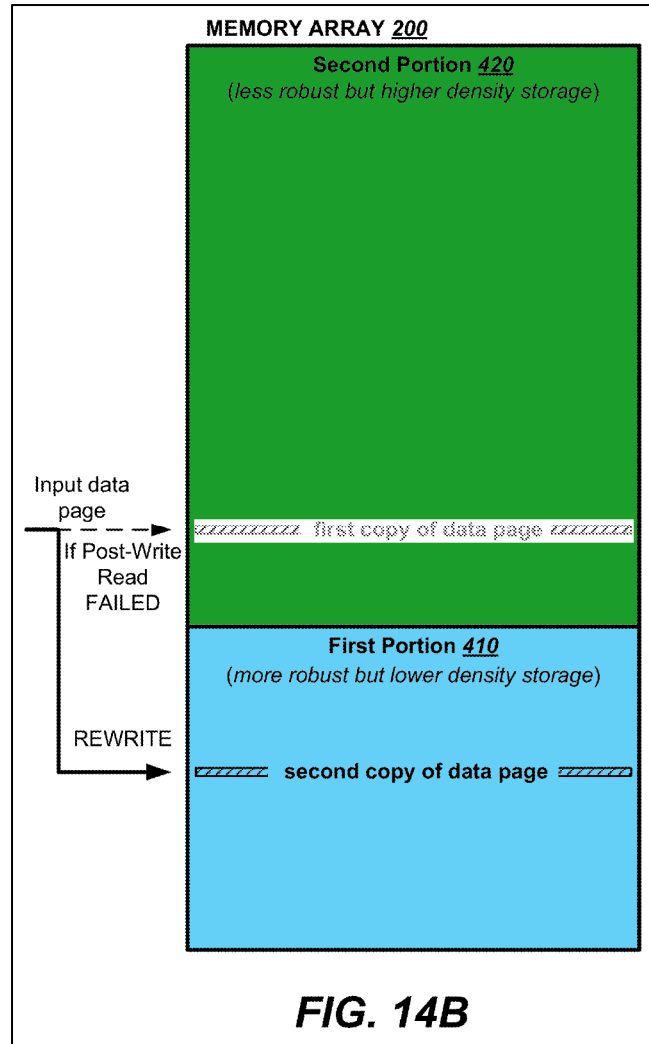
When a page is read from the memory array, it is “shifted out” (or “toggled”) to **controller 102** where ECC can be used to correct any errors in the read data. *Id.*, 12:64-13:8. Gavens teaches that using ECC in this way—where the ECC must be programmed to correct the “worst-case” number of bits—is burdensome in terms of processing time as well as chip area on the controller. Ex. 1045, 3:48-4:2; *see also id.*, 5:15-30, 13:9-16.

To address this problem, Gavens discloses different embodiments of **memory array 200**, including those of Figures 14A-14B and 16A-16C. These embodiments rely on a memory with a first SLC portion (which is less error prone) and a second MLC portion (which is more error prone). Gavens explains that optimizing where data is stored (in the SLC or MLC portion) based on the expected number of error

bits “allows a smaller and more efficient...ECC...to be designed for correcting a smaller number of error bits, thereby improving the performance and reducing the cost of the memory.” *Id.*, 4:28-31.

a. First Embodiment (Figs. 14A-14B)

Gavens describes a first embodiment with respect to Figures 14A-14B. *See* Ex. 1045, 17:24-26. As shown in Figure 14B below, Gavens describes a memory array including a “**lower density**” First Portion 410 (*i.e.*, SLC) and a “**higher density**” Second Portion 420 (*i.e.*, MLC). *Id.*, 4:13-19, 16:14-28. The MLC portion is more error prone than the SLC portion but can be used “for efficient storage.” *Id.*, 4:16-19, 16:41-45. Gavens explains that data can first be written to the MLC portion, then “read back in a post-write read operation to check for excessive error bits.” *Id.*, 4:19-21; *see also id.*, Fig. 14A, 16:41-56; Ex. 2001, ¶¶71-73.



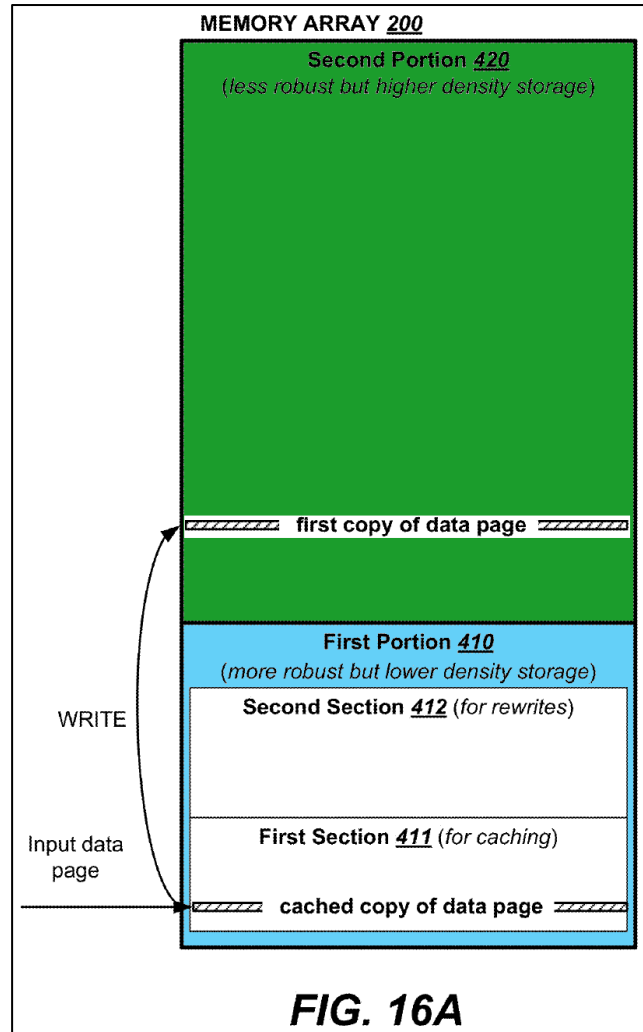
Ex. 1045, Fig. 14B (annotated).

The post-write error detection process can check for errors in one of two ways: “either by comparison with the original copy which may be cached or by checking the...ECC.” *Id.*, 16:46-49. Gavens explains that the “data is cached in [a] first section of the **first portion**” of non-volatile memory (*i.e.*, a section of the non-volatile SLC memory). *Id.*, 18:1-8; *see also id.*, 16:14-16.

Once the error rate is checked, the device determines whether to keep the data in MLC or instead move it to SLC. “If the error bits exceeded a predetermined amount, the data is” stored in “the less error-prone” SLC portion. Ex. 1045, 4:21-22; *see also id.*, Fig., 14B, 16:41-56, 17:7-20, 18:1-8.

b. Second Embodiment (Figs. 16A-16C)

Figures 16A-16C depict a second embodiment of Gavens. *See* Ex. 1045, 7:26-28. For this embodiment, as shown in Figure 16A below, Gavens describes splitting the lower density First Portion (*i.e.*, SLC) into a “first section” 411 “for caching” write data, and a “second section” 412 “for rewrit[ing]” data from the higher density Second Portion (*i.e.*, MLC). *Id.*, Fig. 16A, 18:9-23; Ex. 2001, ¶¶74-79.



Ex. 1045, Fig. 16A (annotated).

Incoming data is written both to the MLC portion and as a “cached copy” to the first section of the SLC portion. *Id.*, 18:21-29. Detecting the number of error bits is then “accomplished by comparison [of the data written to MLC] with the cached copy.” *Id.*, 18:40-44. This comparison may be performed at the **data latches 430** (highlighted orange in Figure 1 above). *Id.*, 16:46-49, 18:30-39. Gavens recommends using **data latches 430** because it eliminates the need to “toggle” data out to **controller 102**, as would be required if using ECC Processor 62. *Id.*, 18:36-

39; *see also id.*, 3:60-4:2. “[M]uch time can be saved” by eliminating the need to toggle data to the **controller 102**, thus increasing the performance of the device. *Id.*

If there is an acceptable number of error bits, the MLC portion becomes the “valid” copy. *Id.*, 18:45-52, Fig. 16B. If there is an unacceptable number of error bits, the “cached copy” in the “first section” of the SLC is rewritten to the “second section” of the SLC. *Id.*, 18:53-67, Fig. 16C.

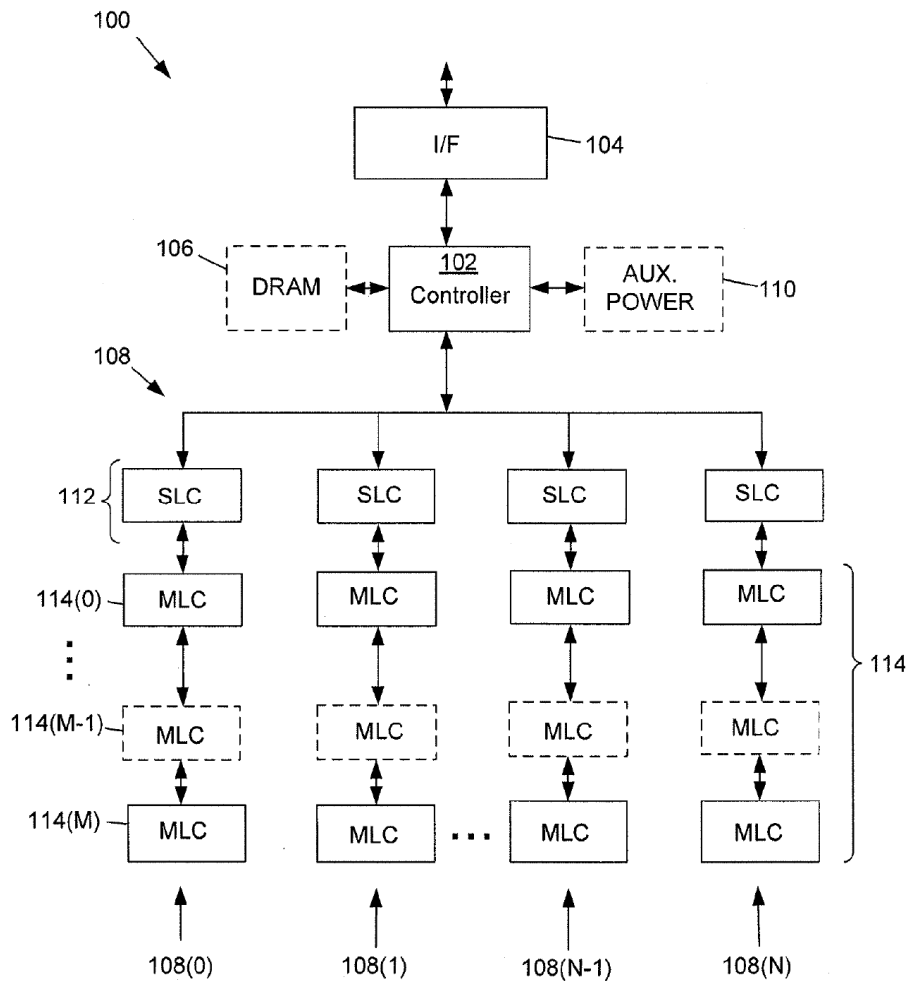
Gavens emphasizes that, by caching the data in the non-volatile memory array and using the latches to compare the data, this approach lowers overhead and limits reliance on **controller 102**. *See* Ex. 1045, Fig. 1, 3:60-4:2, 18:36-67; Ex. 2001, ¶79.

c. Third Embodiment (Fig. 19): “Hot Count”

Gavens presents a third embodiment with Figure 19, where “the age of the memory device is determined by a hot count maintained with each erase block of memory cells.” Ex. 1045, 19:37-20:39, 19:47-49; *see also id.*, 7:30-32. The “hot count” tracks “the number of times each erase block has been cycled through erase and program operations.” *Id.*, 19:47-54. Once the “hot count” for an erase block reaches a threshold, “enhanced post-write-read error management” is enabled. *Id.* Gavens explains, for example, that, when enhanced error management is enabled for an MLC block, the error rate of that block is determined and, if the error rate is excessive, the data from that block may be moved to a different MLC block. *See id.*, 20:30-50; Ex. 2001, ¶80.

B. Moshayedi (Ex. 1043)

As shown in Figure 1 below, Moshayedi describes a flash memory having both SLC and MLC memory, with the described goals of “accommodat[ing] application memory needs at desirable prices, in addition to increasing read/write performance.” Ex. 1043, Abstract, ¶8; Ex. 2001, ¶¶81-85.



Ex. 1043, Fig. 1.

To accomplish these goals, the memory distinguishes between “static” (not frequently written) and dynamic (“frequently written”) data, and changes where that

data is stored—in MLC or SLC—accordingly. *Id.*, Abstract. Moshayedi explains that all data may initially be written to SLC. *See id.*, ¶¶9 (“The threshold may be set at 0 initially, resulting in all data being written to SLC flash...”), 54, 73. But later, the memory may rely on both MLC and SLC, and determines where to store data based on whether it is “static” and “dynamic.” *Id.*, ¶¶9, 24, 31-33. Moshayedi describes several, alternative embodiments for doing this, including embodiments that consider (1) write count, (2) erase count, and (3) error count. *Id.*; Ex. 2001, ¶82.

Write Count: Moshayedi explains that the memory may distinguish between static and dynamic data by determining the **write count** of a logical block address (LBA). *See* Ex. 1043, Abstract, ¶¶9, 24. The memory “determines whether to store ***newly received data*** associated with a particular LBA in SLC flash or in MLC flash depending on the number of writes that have occurred for that particular LBA.” Ex. 1043, ¶¶9, 24. Unlike the Challenged Claims—where the respective contents of blocks are transferred from MLC to SLC—here, only “newly received data” is ***redirected*** from MLC to SLC. Ex. 2001, ¶83.

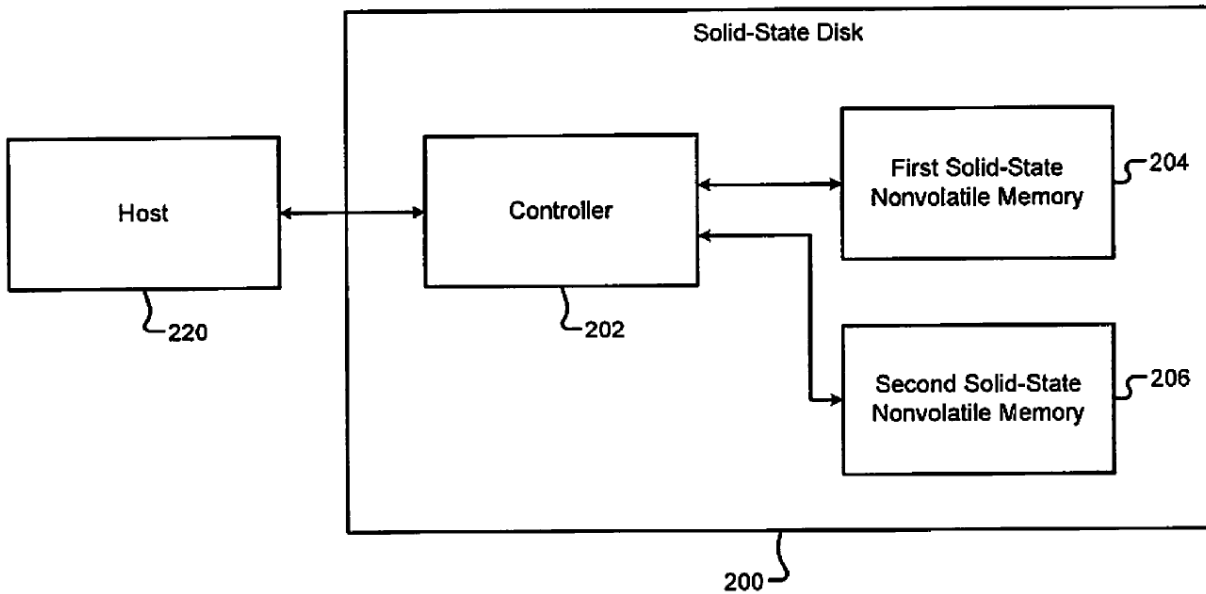
Erase Count: Alternatively, static and dynamic data can be distinguished based on the “erase count” of physical blocks. Ex. 1043, ¶¶30, 32, 47-50. “The flash drive may use the erase count to move data between MLC flash and SLC flash.” *Id.*, ¶32. Moshayedi explains that, “once a block in MLC flash reaches a threshold erase count...the **next write operation to that block triggers a swap** where the data

from the MLC flash block is written to a block in SLC flash.” *Id.* Thus, unlike the Challenged Claims and like the “write count” embodiment described above, only new data—for the “next write operation”—is *redirected* from MLC to SLC. Ex. 2001, ¶84.

Error Count: Moshayedi describes another alternative embodiment, which relies on the error count of physical blocks to determine whether data is static/dynamic. Ex. 1043, ¶¶31, 33. This embodiment describes relocating data blocks with a threshold “number of read errors” to “*data blocks with less wear.*” Ex. 1043, ¶33. Unlike the write count and erase count embodiments, this embodiment does not disclose redirecting data from MLC to SLC; instead, data is redirected to “blocks with less wear.” *Id.*; Ex. 2001, ¶85.

C. Sutardja (Ex. 1042)

Sutardja describes a system with a “First...Memory” 204 and “Second...Memory” 206, as depicted in Figure 2 below. Sutardja describes mapping logical addresses to the first and second memories, based on characteristics of each memory related to their projected lifetimes. Ex. 1042, Abstract; Ex. 2001, ¶¶86-89.



Ex. 1042, Fig. 2.

Particularly, Sutardja describes performing “degradation testing” on physical addresses to determine the storage capability of the memory. Ex. 1042, ¶¶134-139, 151. Sutardja explains that, during “periods of inactivity,” *test data* can be written to a physical address. *Id.*, ¶135 (“degradation testing module...*may provide addresses and data*”). The test data is read back to check for errors and, based on that test, a “degradation value for the address” is determined. *Id.*, ¶¶135-138. The memory “may adapt its mapping based on the degradation value.” *Id.*, ¶138. For example, if one of the first or second memories (204 and 206 in Figure 2 above) fails the degradation test, the memory “may assign *all new writes* to the other one of the memories 204 and 206” that did not fail the test. *Id.*, ¶139. Accordingly, unlike the Challenged Claims, and just like Moshayedi (explained above), Sutardja does not

disclose transferring the respective contents of blocks based on the degradation testing—only “assign[ing]”/redirecting “*new writes*.” *Id.*; Ex. 2001, ¶87.

Sutardja describes additional embodiments where metrics about the memory are determined and, based on those metrics, the memory adjusts its mapping. *First*, with Figure 7A and ¶¶146-147, Sutardja considers the “write frequencies for *logical addresses* where data is *to be written*” and adjusts its mapping to the first/second memory accordingly. *Second*, with Figure 7C and ¶149, Sutardja considers “the number of write operations to [a] first [*physical*] *block*...during a predetermined time” and adjusts its mapping accordingly. *Third*, with Figure 7E and ¶153, Sutardja describes mapping data based on whether “the wear level of the first...memory is greater than a predetermined threshold”—if so, the controller “maps *all* the logical blocks to physical blocks of the second...memory.” *See* Ex. 2001, ¶88.

VII. REASONS FOR DENYING INSTITUTION

The Board should deny institution because the Petition fails to establish a reasonable likelihood that it would prevail on any of its three grounds.

A. Ground 1: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Gavens

As discussed below, the Petition’s analysis for Ground 1 (obviousness over Gavens) is fundamentally flawed with respect to multiple limitations of Claim 1 of the Challenged Patent, and those flaws propagate to each challenged dependent claim. Each of the below flaws by itself warrants denial of institution.

1. [1b]: “SLC non-volatile memory module comprising a plurality of *individually addressable blocks*”

For limitation [1b], Petitioner fails to explain how Gavens teaches “individually addressable blocks.” Petition, 24; Ex. 2001, ¶91. Notably, the Petition’s discussion of [1b] fails to mention “individually addressable blocks” altogether.

Thus, Petitioner failed to meet its burden to set forth its mapping to Gavens clearly and with particularity. 35 U.S.C. § 312(a)(3); *GN Resound A/S v. Oticon A/S*, IPR2015-00103, Paper 13, 6 (P.T.A.B. Jun. 18, 2015) (“It is a requirement of a Petition to align the evidence and arguments with the various limitations of the challenged claims.”). This deficiency is incurable, and Ground 1 fails on this basis alone. *See Nearmap US, Inc. v. Eagle View Technologies, Inc.*, IPR2024-00716, Paper 9, 18 (P.T.A.B. Oct. 9, 2024) (denying institution, explaining that “[t]he question before us is not whether we could come up with a way to understand how the asserted prior art could render the challenge claims unpatentable, but rather did Petitioner provide the arguments and evidence to show that the challenged claims are unpatentable”).

2. [1c]: “a controller coupled to” an “MLC non-volatile memory module” and an “SLC non-volatile memory module”

For limitation [1c], Petitioner’s obviousness analysis is deficient in multiple respects.

First, for this limitation, Petitioner relies only on its own construction for “SLC non-volatile memory”: “non-volatile memory cells arranged with circuitry incapable of storing multiple logical pages in a single physical page of cells.” Petition, 20. Applying this construction, Petitioner suggests that limitation [1c] requires a “‘*native*’ SLC flash chip.” Petition, 25. While PO disagrees with this construction, even if it is applied—where Petitioner presents its ground based solely on its own construction—Petitioner’s ground fails.³

Specifically, Petitioner fails to explain why it would have been obvious to replace one of Gavens’s MLC memory chips—which are each coupled to Gavens’s “controller 102”—with a “‘*native*’ SLC flash chip.” Petition, 25; *see* Ex. 2001, ¶¶92-94. Instead, Petitioner merely asserts that this modification would have been obvious, and cites Gavens at 8:17-18 and 8:25-29. *See* Petition, 25 (“It would be obvious to have one or more of those NAND flash chips be ‘*native*’ SLC flash chips.... Gavens...disclose[s] the controller coupled to an MLC module and obviously substituted SLC module...”). These citations, however, do not support Petitioner’s conclusory argument. The cited portions of Gavens say nothing about

³ Thus, the Board need not resolve the parties’ dispute over the proper construction of “SLC non-volatile memory” to deny the Petition. *See also supra* Section V (Claim Construction).

SLC, let alone “native” SLC, as Petitioner contends. These citations describe only Gavens’s “memory device” and its controllers. Ex. 1045, 8:17-18 (“The memory device 90 includes one or more memory chip 100 managed by a controller.”), 8:25-29 (“The control circuitry 110 is an on-chip controller.... The control circuitry 110 typically....”). The paragraph of Gavens that Petitioner cites instead directly contradicts its argument, stating that “memory chip 100 includes...cell[s] capable of being configured as a *multi-level cell (‘MLC’) for storing multiple bits of data.*” *Id.*, 8:18-21 (emphasis added). Thus, Petitioner fails to support its assertion that a POSA would have “substituted” one of Gavens’s MLC chips with a “‘native’ SLC flash chip[.]” Ex. 2001, ¶94.

Accordingly, Petitioner fails to demonstrate why Gavens renders obvious using a “‘native’ SLC flash chip.” “[O]bviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006); *see also* 37 C.F.R. § 42.22(a)(2) (requiring an “explanation of the significance of the evidence”). Notwithstanding, it is not PO’s duty—or the Board’s—to parse through Petitioner’s citations to guess how Petitioner is supporting its conclusory assertion of obviousness; Petitioner has the duty to set forth its ground clearly and with particularity and has failed to do so. 35 U.S.C. § 312(a)(3); *see also DeSilva v. DiLeonardi*, 181 F.3d 865, 866-67 (Fed. Cir.

1999) (“A brief must make all arguments accessible to the judges, rather than ask them to play archeologist with the record.”).

Relatedly, Petitioner’s obviousness analysis is also deficient because Petitioner fails to explain why a POSA would have been motivated to make this modification or establish that a POSA would have had a reasonable expectation of success in doing so. The Petition entirely omits these essential elements from its obviousness analysis. “[T]here must be a motivation to make the combination and a reasonable expectation that such a combination would be successful, otherwise a skilled artisan would not arrive at the claimed combination.” *In re Stepan Co.*, 868 F.3d 1342, 1346 n.1 (Fed. Cir. 2017).

Second, contrary to Petitioner’s conclusory assertion, a POSA would not have been motivated to replace one of Gavens’s memory chips with a “‘native’ SLC flash chip.” *See* Ex. 2001, ¶95. Gavens explains that its memory chips are already capable of operating with one portion as MLC, and another portion as SLC (albeit, not “native” SLC). *See* Ex. 1045, Abstract, 4:13-31, 6:17-25, 15:62-16:12, 16:17-25. That is the essence of Gavens’s invention—taking advantage of the efficiencies of MLC storage, by relying on a “less error-prone” SLC *portion* when an MLC portion has too any errors. *See id.*, 4:13-31, 4:18-22 (“Data is written to the second portion for efficient storage. Afterwards, the data is readback in a post-write read operation to check for excessive error bits. If the error bits exceeded a predetermined amount,

the data is rewritten or kept at the less error-prone first portion.”), 15:66-16:13, 16:41-45; *see also* Section VI.A (describing Gavens).

Additionally, Gavens explains that using the SLC portion to cache data, as described in its first and second embodiments (*see supra* Section VI.A), contributes to avoiding the need to “toggle[]” data out to controller 102 when comparing data for error checking. *See* Ex. 1045, 18:21-39, 18:36-39. This is because “data latches” within the memory chip (which includes both the MLC and SLC portions) enable data comparison without relying on external controller 102. *See id.*, 18:30-39, Fig. 1 (showing “Data Latches” 430 as part of the Memory Chip, separate from the Controller 102). Replacing the SLC portion of the *same* memory chip with a *separate* SLC memory chip connected to the controller 102 would negate this benefit, as data cached in a separate SLC chip would need to be toggled out for comparison with data stored in the MLC chip.

Therefore, Gavens teaches away from Petitioner’s proposed modification, and a POSA would not have been motivated to replace one of Gavens’s MLC memory chips with a “‘native’ SLC flash chip,” as this modification would undermine Gavens’s intended advantage—efficiently utilizing higher-density MLC memory. *See* Ex. 2001, ¶96. Petitioner’s proposed modification “would require a substantial reconstruction and redesign of the elements shown in [Gavens] as well as a change in the basic principles under which [Gavens] was designed to operate” and cannot

be correct. *Samsung Elecs. Co., Ltd. v. Mojo Mobility Inc.*, IPR2023-01099, 2024 WL 457731, at *10 (P.T.A.B. Feb. 6, 2024) (denying institution) (quoting *Application of Ratti*, 270 F.2d 810, 813 (C.C.P.A. 1959)).

3. [1f]: “c) determine which of the blocks of the plurality of blocks...are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed”

Petitioner fails to identify with particularity which prior art disclosure it is mapping to this limitation, particularly with respect to “maintaining a count.” Petition, 29-30. Instead, Petitioner takes a scattershot approach, ambiguously referencing multiple unrelated disclosures that mention a “count,” without explaining how they are mapped—individually or in combination—to the claims. *See* Ex. 2001, ¶¶98-99. Petitioner first cites Gavens’s “hot count” but then also relies on Lee (Ex. 1050) and Gorobets (Ex. 1049), which Gavens allegedly incorporates by reference, citing their unrelated teachings on maintaining “counts” for memories. Petition, 30. However, Petitioner provides no explanation of how Lee and/or Gorobets—which do not describe the same embodiments as Gavens—apply to or combine with Gavens’s disclosure.⁴ Petitioner cannot simply cherry-pick distinct

⁴ Instead, Petitioner provides only an anticipation analysis, alleging that “Gavens and incorporated references *disclose* the determining [the count].” Petition, 30. There is no obviousness analysis for this limitation in the Petition. *Id.*

teachings from different references without articulating how they are mapped to the claims. *See Captioncall, LLC v. Ultratec, Inc.*, No. IPR2020-01215, 2021 WL 278381, at *7 (P.T.A.B. Jan. 27, 2021) (denying institution when petitioner relied on “distinct” embodiments from a reference and another reference that was incorporated by reference, explaining that “[a]nticipation requires all elements arranged as in the claim without the need to pick and choose elements from different disclosures”).

Additionally, Petitioner relies on Gavens’s third (Figure 19) embodiment for this limitation but relied on Gavens’s first (Figure 14) and second (Figure 16) embodiments for limitation [1e]. *See supra* Section VI.A (describing Gavens’s embodiments). However, Petitioner fails to provide the required obviousness analysis to connect its arguments across multiple embodiments, instead merely citing between these embodiments without explaining why a POSA would have been motivated to combine them. Specifically, Petitioner cites Figure 19 for its discussion of the “hot count.” Petition, 29-39; *see* Ex. 1045, 19:47-49, 20:7-29 (describing Figure 19 as part of “a preferred embodiment”); *see also* Section VI.A.c (describing Gavens’s third embodiment); Petition, 30-31 (again relying on Figure 19 for limitation [1g]). But Petitioner relied on Figures 14 and 16 for limitation [1e]. *See* Petition, 27-28; *see* Ex. 1045, 16:14-15 (referring to “Fig. 14A” as part of “a preferred embodiment”), 18:9-12 (referring to “Fig. 16A” as part of another

“preferred embodiment”); *see also* Sections VI.A.a-VI.A.b (describing Gavens’s first and second embodiments).

Petitioner cannot mix and match Gavens’s embodiments without explaining how and why those embodiments would have been combined. “Petitioner’s challenge [incorrectly] treats the various features of [Gavens’s] different embodiments as if they are interchangeable with one another.” *Abiomed, Inc. v. Maquet Cardiovascular, LLC*, IPR2017-01204, 2017 WL 4804523, at *4 (P.T.A.B. Oct. 23, 2017) (denying institution). For limitation [1f], Petitioner provided only an anticipation analysis, and did not even attempt to provide an obviousness rationale for combining Gavens’s embodiments. *See* Petition, 30 (“Gavens and incorporated references *disclose*...”).

Petitioner’s scattershot approach—citing disparate embodiments from Gavens, Lee, and Gorobets, without clearly identifying which *one* Petitioner is relying on—must be rejected. *See Masimo Corp. v. Apple Inc.*, IPR2023-00745, Paper 44 (final written decision), at 36 (P.T.A.B. Oct. 15, 2024) (determining that the challenged claim is not unpatentable, explaining: “In essence, Petitioner invites the Board to piece together a single theory that meets all claim elements from these disparate theories that each require distinct structures...”).

4. [1f]: “*controller adapted to*” ([1c]) ... “maintain[] a count of the number of times each of the blocks is accessed”

For limitation [1f], Petitioner also fails to demonstrate that Gavens’s Figure 19 discloses “*controller 102*” maintaining the “hot count.” Notably, limitation [1c] requires a “controller adapted to” perform limitations [1d]-[1g]. For limitation [1c], Petitioner maps the claimed controller to Gavens’s “controller 102.” Petition, 25. Therefore, for limitation [1f]—as well as limitations [1d]-[e] and [1g]—Petitioner must show that the same controller 102 performs each claimed function. Petitioner fails to do so. *See* Ex. 2001, ¶¶100-103.

Instead, for limitation [1f], Petitioner conclusorily states, “Although (and because) *the controller uses* that information to determine according to a threshold count whether a particular post-write-read scheme should be implemented..., *the controller clearly* is adapted to maintain the counts and determine from those counts the blocks accessed most frequently.” Petition, 29-30. But Petitioner provides no explanation how Gavens discloses that *controller 102* performs these functions. Petitioner cites Gavens at 20:18-27 and Figure 19, but this disclosure says nothing about controller 102, or any other controller.

To the contrary, a POSA would have understood that a different and separate controller—“*on-chip control circuit 110*”—maintains Gavens’s “hot count”:

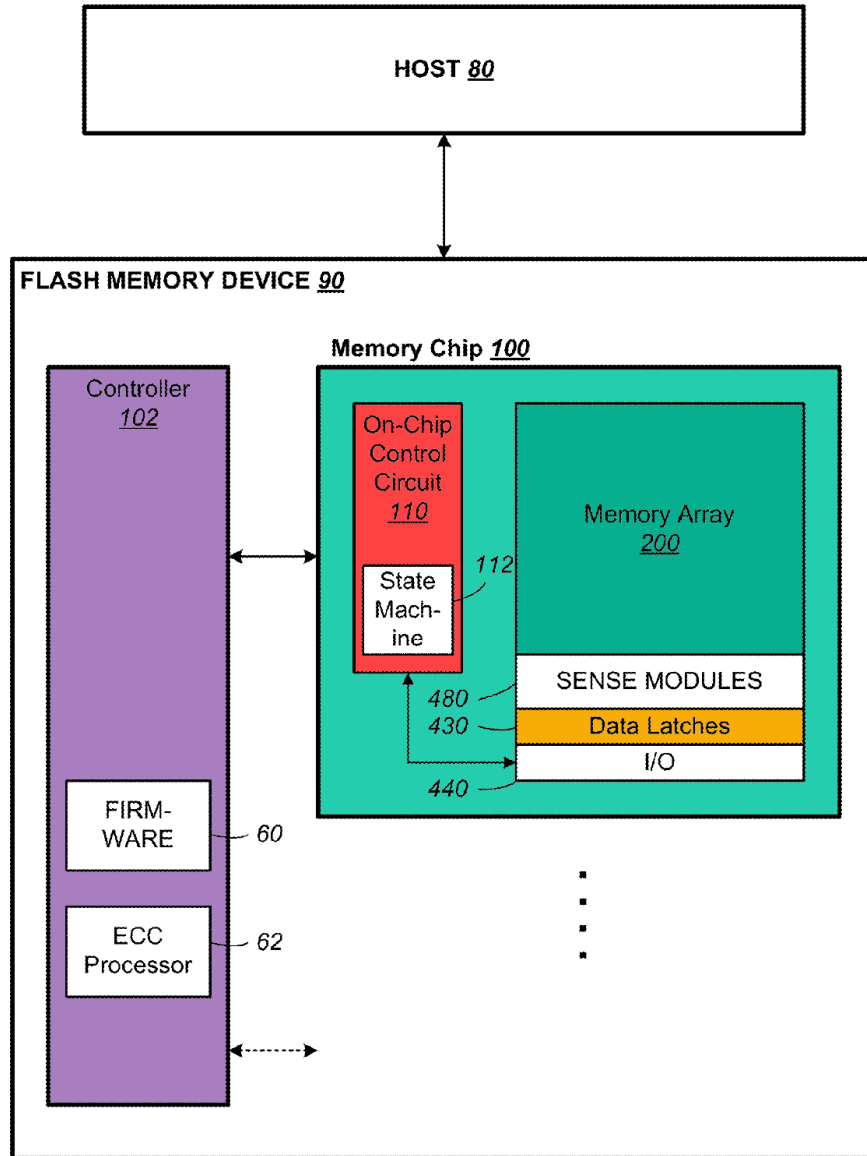


FIG. 1

Ex. 1045, Fig. 1 (annotated); *see* Ex. 2001, ¶102. Gavens explains that “the hot count for each block can be stored in the system data area of the **data page 70**....”

Ex. 1045, 20:2-6; *see also id.*, 12:66-67 (“The **data page**...is then written to the **memory array 200**.”). Gavens does not specify which controller maintains the “hot count,” and its storage in the **memory array 200**. A POSA would have understood,

however, that this function would be performed by **on-chip control circuit 110**—not **controller 102**, as required for Petitioner to remain consistent with its analysis for limitation [1c]. Gavens distinguishes between these components, explaining that “on-chip control circuitry 110 controls low-level memory operations of each chip,” while “controller 102...controls and manages higher level memory operations.” Ex. 1045, 8:23-25, 8:32-33. Accordingly, a POSA would have determined that on-chip control circuit 110 maintains the hot count, as this qualifies as a “low-level memory operation.” See Ex. 2001, ¶103.

Petitioner, in contrast, *provides no obviousness analysis as to why a POSA would have understood controller 102 to maintain the hot count*. Petition, 29-30 (alleging only that “Gavens...disclose[s]”). As explained above, Gavens itself does not disclose this, nor would it have been obvious.

5. [1g]: “allocate those blocks that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module”

Petitioner contends that Gavens discloses limitation [1g] under both of its constructions of “blocks”: logical and physical. Petition, 30-32. Both arguments fail.⁵ Ex. 2001, ¶¶104-107.

⁵ Accordingly, the Board need not resolve the underlying dispute regarding the proper construction of “blocks” to reject this ground. See *supra* Section V.A.

Physical blocks: As explained above, “blocks” should be construed to refer only to “physical” blocks. *See supra* Section V.A. Applying this construction, Petitioner’s mapping is deficient. *See* Petition, 30-31. Petitioner does not explain how Gavens teaches “transferring” physical blocks to SLC, as required by limitation [1g].

Petitioner merely asserts that Gavens discloses that “when blocks exceeding the threshold (receiving the most frequent writes) are written with data that fails a data integrity test, *they are rewritten (transferred) to SLC.*” Petition, 30-31. However, Petitioner provides no supporting evidence for this assertion. While citing Gavens’s Figure 19 embodiment, including Gavens at 20:21-29, Petitioner fails to show that this passage addresses data transfer in any capacity. *Contra* Ex. 1045, 20:21-29 (stating only that “error management for the rest of the life of the memory” is “enable[d]” when a “memory block” fails the data integrity test).

Petitioner also cites other portions of its Petition and references that Gavens allegedly incorporates by reference, but these fail to compensate for the Petition’s lack of evidentiary support. *First*, Petitioner references “element 1[e],” yet its analysis for that limitation relies on the distinct embodiments of Figures 14 and 16—not Figure 19, which Petitioner relies on for limitation [1g]. *See* Petition, 27-29; *see also supra* Section VI.A (explaining Gavens’s embodiments). *Second*, Petitioner cites Lee and Gorobets, which Gavens allegedly incorporate by reference, but as

explained above, Petitioner cannot merely cite disparate embodiments from those references without providing the required obviousness analysis.

Thus, “[i]t is the petitioner’s burden to present a clear argument,” and it has failed to do so. *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023). “It is of the utmost importance that petitioners in the IPR proceedings adhere to the requirement that the initial petition identify ‘with particularity’ the ‘evidence that supports the grounds for the challenge to each claim.’” *Intelligent Bio-Sys., Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016) (quoting 35 U.S.C. § 312(a)(3)).

Logical blocks: Petitioners ground also fails under its construction of “logical” blocks. For this, Petitioner relies entirely on teachings from secondary references, Sutardja (Ex. 1042) and Moshayedi (Ex. 1043). Petition, 31-32. However, its analysis is plainly deficient. Petitioner cites teachings from these unrelated references but fails to provide the requisite obviousness analysis, including motivation to combine and reasonable expectation of success.⁶ Petitioner offers none of this, and its ground must be rejected. *See In re Stepan*, 868 F.3d at 1346 n.1.

⁶ Petitioner does not even attempt to invoke obviousness, instead stating that “Gavens...in view of...Sutardja, Moshayedi...*disclose*...[1g].” Petition, 32.

Moreover, the alleged teachings of this limitation in Sutardja and Moshayedi are also deficient, at least for the reasons discussed below for Grounds 2 and 3.

B. Ground 2: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Moshayedi

As discussed below, the Petition’s analysis for Ground 2 (obviousness over Moshayedi) is fundamentally flawed with respect to multiple limitations of Claim 1 of the Challenged Patent, and those flaws propagate to each challenged dependent claim. Each of the below flaws by itself warrants denial of institution.

1. [1b]: “at least one SLC non-volatile memory module comprising a plurality of *individually erasable blocks*”

Petitioner provides only a cursory, single-sentence analysis for this limitation, entirely omitting any discussion of the claimed “individually erasable blocks.” Petition, 40. Thus, the Petition’s deficient Ground 2 analysis should be rejected for this reason alone. *See Nearmap*, IPR2024-00716, Paper 9, 18.

Further, for this limitation, Petitioner supports its Ground with a single citation to Moshayedi. Petition, 40. However, even that citation cannot save Petitioner’s ground. Petitioner cites “Fig. 1(112),” but this figure says nothing about “individually erasable blocks” and, instead, depicts only SLC. Ex. 1043, Fig. 1; Ex. 2001, ¶109.

Finally, Petitioner cites two paragraphs of its expert declaration but fails to provide any analysis from those paragraphs in the Petition itself. Petition, 40 (“See

[Sechen ¶¶167-68]”). Petitioner cannot simply cite its expert declaration while omitting arguments/analysis from the Petition regarding Petitioner’s mapping of prior art to claim language. 37 C.F.R. § 42.6(a)(3); *Acon Inc. v. Amo Dev., LLC*, IPR2021-00858, 2022 WL 17585777, at *16 (P.T.A.B. Dec. 12, 2022) (“arguments may not be incorporated from an expert’s declaration wholesale into the [p]etition”).

Accordingly, Petitioner has not met its burden to clearly map limitation [1b].
See 35 U.S.C. § 312(a)(3).

2. **[1d]: “a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a list of logical address ranges accessible by a computer system, the list of logical address ranges having a *minimum quanta of addresses,...*”**

For limitation [1d], Petitioner fails to identify how Moshayedi discloses a “list of logical address ranges having a *minimum quanta of addresses.*” Petition, 41; Ex. 2001, ¶¶110-111.

Petitioner references several paragraphs from Moshayedi but fails to correlate their teachings with the claim language. Petition, 41 (citing Ex. 1043, ¶¶6, 36-39, 44). Petitioner cannot simply cite Moshayedi without explaining how it discloses what is claimed. *See DeSilva*, 181 F.3d at 866-67.

Still, even when considered individually, these citations do not show that Moshayedi discloses or suggests the “minimum quanta of addresses.” Paragraphs 6 and 44 disclose logical-to-physical mapping but do not describe a minimum quanta

of addresses. *See* Ex. 1043, ¶¶6 (“logical block address (‘LBA’”), 44 (“assigning LBA to physical address”). Similarly, paragraphs 36-39 disclose a “controller architecture” and “virtual-to-physical mapping” but do not describe a minimum quanta of addresses. *See id.*

Accordingly, Petitioner fails to map this limitation with particularity, and its ambiguous Petition should be rejected. *See* 35 U.S.C. § 312(a)(3).

3. [1e]: “b) determine if a range of addresses listed by an entry and mapped to a similar range of physical addresses within the at least one *MLC non-volatile memory module*, fails a data integrity test, and, in the event of such a failure, the controller *remaps the entry to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module*”

Petitioner fails to show that Moshayedi discloses or suggests that “the controller remaps” a range of logical addresses that “fails a data integrity test” from MLC to SLC, as required by [1e]. Petition, 42-43; Ex. 2001, ¶¶112-116.

First, Petitioner relies on ¶¶31 and 33 of Moshayedi, which describe relocating data blocks with a threshold “number of read errors” to “data blocks with less wear.” Ex. 1043, ¶33. Based on this, Petitioner makes two assumptions: (1) that a POSA would have understood “‘data blocks with less wear’ to mean SLC”; and (2) that the data blocks would be relocated *from MLC*. Petition, 42-43. Both assumptions are incorrect. Ex. 2001, ¶113.

When describing other embodiments, Moshayedi explicitly refers to relocating data between MLC and SLC, and does not use the term “blocks with less wear.” For example, Moshayedi clearly describes “mov[ing] data between MLC flash and SLC flash” based on erase count—distinct from the error count referenced in ¶¶31 and 33. Ex. 1043, ¶32 (“The flash drive may use the erase count to move data between MLC flash and SLC flash.”). In ¶33, however, Moshayedi uses different language—“blocks with less wear”—which a POSA would not have understood to be a reference to SLC, as Petitioner contends.

Moshayedi instead suggests that SLC blocks could have *more* wear, not less. Moshayedi explains that *data is initially written only to SLC*. See, e.g., Ex. 1043, ¶¶9 (“The threshold may be set at 0 initially, resulting in all data being written to SLC flash...”), 24 (same), 54 (“The amount of SLC that is designated to fill-up initially, before factoring write frequency and moving blocks to MLC, can be any number (%).”). If data is written only to SLC and no data has been written to MLC, then MLC blocks would have “less wear.” Ex. 2001, ¶115. Certainly, Moshayedi at ¶33 does not disclose the remapping from MLC to SLC required by the claim limitation merely by disclosing a “relocat[ion] to data blocks with less wear.”

Second, Petitioner provides an “[a]lternative” mapping based on a secondary reference, Gavens—which is not even a reference in the ground. Petitioner, 43; see also *id.*, 17 (specifying that Ground 2 only relies on Moshayedi). However, once

again, Petitioner fails to present any obviousness analysis for this ground. *Id.*, 43. Petitioner, for example, does not follow a *Graham* analysis to identify the teachings of the primary reference and what aspect of the claim language is missing from it such that it would be obvious to implement those features, and fails to provide any motivation to combine Moshayedi and Gavens' disparate teachings. This "alternative" mapping should also be rejected. *See In re Stepan Co.*, 868 F.3d at 1346 n.1.

4. [1g]: "d) allocate those blocks that receive the most frequent writes by *transferring the respective contents of those blocks* to the at least one SLC non-volatile memory module"

In a single sentence, Petitioner asserts that Moshayedi discloses limitation [1g] under both of its constructions of "blocks": logical and physical. *See* Petition, 45.⁷ Both arguments fail. Ex. 2001, ¶¶117-125

Physical blocks: Petitioner does not and cannot show that Moshayedi discloses "transferring the respective contents of those blocks" that "receive the most frequent writes" to SLC, as required by [1g].

For this ground, the Petition simply states, "in another embodiment, the physical blocks are counted," followed by a block quote and a list of citations to

⁷ The Petition in no way indicates that it is relying on an obviousness analysis for limitation [1g].

Moshayedi. *Id.* (citing Ex. 1043, ¶¶30, 32, 47-50, 60; Figs. 7A, 8). Petitioner fails to explain how this “count” results in “transferring the respective contents of...blocks” to SLC. Instead, Petitioner again leaves it to PO and the Board to try to infer its argument from its citations. This ground should be rejected on that basis alone. *See GN Resound A/S*, IPR2015-00103, Paper 13, 6 (denying institution on certain grounds, explaining that “[e]ssentially, Petitioner requires the Board to search through the cited portions of [the prior art] to map prior art disclosure with claim elements. It is a requirement of a Petition to align the evidence and arguments with the various limitations of the challenged claims.”); *A.C. Dispensing Equip. Inc. v. Prince Castle LLC*, IPR2014-00511, Paper 6 at 5-6 (P.T.A.B. Sept. 10, 2014) (“Petitioner should not expect the Board to search the record and piece together the evidence necessary to support Petitioner’s arguments.”).

Still, even after searching through Petitioner’s citations, it is clear that Moshayedi does not disclose or suggest “transferring the respective contents” of frequently written MLC blocks to SLC; only ***newly received data*** for a next/separate write operation is written to SLC. Moshayedi clearly states in paragraph 32 (cited by Petitioner) that “once a block in MLC flash reaches a threshold erase count...the ***next write operation to that block triggers a swap*** where the data from the MLC flash block is written to a block in SLC flash.” Ex. 1043, ¶32. Paragraph 49 (also cited Petitioner) crystallizes this point, again making clear that, “When the erase

count of the block at MLC area is 500, *next written data* ca[n] require the data to move to SLC area where erase count is less than 500.” Ex. 1043, ¶49; Ex. 2001, ¶119.

Accordingly, unlike [1g], the “respective contents” of the MLC block are never transferred to a SLC block. Rather, during the *next write operation*, a single block of data that is supposed to be written to the MLC block is instead redirected to a SLC block. Ex. 2001, ¶120.

Logical blocks: Petitioner’s ground based on its logical blocks construction fails because, as explained above, “blocks” should be construed to refer only to “physical” blocks. *See supra* Section V.A.

Even applying Petitioner’s incorrect “logical” blocks construction, however, Petitioner cannot show that Moshayedi discloses “transferring the respective contents of those blocks” to SLC, as required by [1g].

Here, Petitioner relies on the write count embodiment of Moshayedi. Petition, 45 (citing Ex. 1043, ¶¶9, 24). Moshayedi explains that this embodiment, just like the “erase count” embodiment of ¶32 (explained above for the “physical” blocks construction), does not result in the transfer of the “contents” of MLC blocks to SLC. Instead, only “*newly received data*” is redirected to SLC. *See, e.g.*, Ex. 1043, ¶9 (“determines whether to store *newly received* data associated with a particular LBA in SLC flash or in MLC flash depending on the number of writes”), 24 (same).

Moreover, Moshayedi explains that the “contents” of MLC blocks are never transferred to SLC blocks, because only the *opposite* transfer is necessary—from SLC to MLC:

As the level of user blocks within the SLC goes above the configured limit, the logical blocks with the lowest write count are moved from SLC to MLC. *No converse operation is necessary* (copy from MLC to SLC) as logical blocks are *directed towards SLC when they are written* by the host if the write count is at a sufficient level to warrant this.

Ex. 1043, ¶73. Moshayedi explains that, while data can be “moved from SLC to MLC,” data is not moved from MLC to SLC. *Id.* Instead, data is only “directed towards SLC *when they are written.*” *Id.*; Ex. 2001, ¶¶121-123.

Physical or Logical Blocks: Under either construction, Petitioner’s ground also fails because Petitioner cites between different embodiments of Moshayedi without providing the necessary obviousness analysis.

For limitation [1e], Petitioner relies on the “read error” embodiment. Petition, 42-43 (citing Ex. 1043, ¶¶31/33). Yet for limitation [1g], Petitioner shifts to the erase count embodiment (*i.e.*, Ex. 1043, ¶32) for the “physical” construction, and the write count embodiment (*i.e.*, Ex. 1043, ¶¶9/24) for the “logical” construction. Petition, 45.

A POSA would have understood these embodiments as *alternative* measures for determining whether data is “static” or “dynamic,” with overlapping functionality in wear leveling blocks. *Compare, e.g.,* Ex. 1043, ¶9 (“frequently written data, which results in frequently erased blocks, in SLC flash, and relatively static data in MLC flash”) *with id.*, ¶30 (“data...is presumed to be more static since the block containing the data has a low erase count”) *with id.*, ¶33 (“Information regarding the number of data read errors associated with a given data block may be used to determine whether the data stored therein is dynamic or static.”); Ex. 2001, ¶¶124-125. Yet, the Petition fails to explain why it would have been obvious to use these embodiments *together*, and Petitioner’s arguments should be rejected. *See Abiomed*, 2017 WL 4804523, at *4.

C. Ground 3: The Petition Does Not Establish That Claim 1 Would Have Been Obvious Over Sutardja

As discussed below, the Petition’s analysis for Ground 3 (obviousness over Sutardja) is fundamentally flawed with respect to multiple limitations of Claim 1 of the Challenged Patent, and those flaws propagate to each challenged dependent claim. Each of the below flaws by itself warrants denial of institution.

1. [1b]: “at least one SLC non-volatile memory module comprising a plurality of *individually erasable blocks*”

Petitioner fails to map limitation [1b] with particularity, omitting any discussion of the claimed “individually erasable blocks.” Petition, 53. Ground 3 should be denied on this basis alone. *See Nearmap*, IPR2024-00716, Paper 9, 18.

Furthermore, Petitioner’s citations cannot save this ground. Petitioner cites “Paragraph [106]” of Sutardja, but this paragraph discloses only “nonvolatile memory” and says nothing about “individually erasable blocks.” Ex. 1042, ¶106; Ex. 2001, ¶¶127-128. Petitioner also cites a paragraph from its expert declaration (“[Sechen ¶235]”) but fails to provide any analysis from that paragraph in the Petition itself. Petition, 53. *See Alcon*, 2022 WL 17585777, at *16 (“[A]rguments may not be incorporated from an expert’s declaration wholesale into the Petition.”). Thus, Petitioner’s limitation [1b] analysis is deficient.

2. [1d]: “a) maintain an address map of at least one of the MLC and SLC non-volatile memory modules, the address map comprising a list of logical address ranges accessible by a computer system, the list of logical address ranges having a *minimum quanta of addresses*,...”

For limitation [1d], Petitioner fails to identify how Sutardja discloses a “list of logical address ranges having a *minimum quanta of addresses*.” Petition, 54-55; Ex. 2001, ¶129. The Petition ignores this limitation entirely. *See Nearmap*, IPR2024-00716, Paper 9, 18.

In any event, Petitioner’s cited portions of Sutardja do not disclose the “minimum quanta of addresses.” Petitioner points to the Abstract and ¶107 of Sutardja, but neither discloses or suggests a “minimum quanta.” Ex. 2001, ¶130; *see* Petition, 54 (vaguely asserting that the abstract and ¶107 disclose a “basic NAND flash system of L2P mapping”). Instead, these portions merely reference mapping “logical addresses” in general terms. Ex. 2042, abstract, ¶107.

Petitioner also cites three paragraphs of its expert declaration (“Sechen ¶¶32-34”) but fails to provide any analysis from these paragraphs in the Petition itself. Petition, 54. Petitioner again ignores that “arguments may not be incorporated from an expert’s declaration wholesale into the Petition.” *Alcon*, 2022 WL 17585777, at *16. In any case, ¶¶32-34 of Dr. Sechen’s Declaration also fail to address the “minimum quanta of addresses.”

3. [1e]: “b) determine if *a range of addresses listed by an entry and mapped to a similar range of physical addresses within the at least one MLC non-volatile memory module, fails a data integrity test, and, in the event of such a failure, the controller remaps the entry to the next available equivalent range of physical addresses within the at least one SLC non-volatile memory module*”

Petitioner fails to show that Sutardja discloses or suggests “remap[ping]” an “entry” that “fails a data integrity test...[from MLC] to...SLC.” Petition, 55-56; Ex. 2001, ¶¶131-135.

For limitation [1e], Petitioner relies on ¶¶134-139 and ¶151 of Sutardja, which describe “degradation testing” of a nonvolatile memory during “periods of inactivity.” Ex. 1042, ¶135. A “degradation testing module” “provide[s] addresses and data”—that is, test data—to the memory. *Id.* The degradation module then performs a data integrity test *on that test data*; specifically, the module “may determine a degradation value for [a tested] physical address” and a “wear leveling module...may adapt its mapping based on the degradation value measured[.]” *Id.*, ¶¶135-138. For example, Sutardja explains that if the degradation value indicates that a memory “is approaching the end of its useful lifetime,” “the wear leveling module...may assign *all new writes* to [a different] memor[y].” *Id.*, ¶139; *see also id.*, ¶151; Ex. 2001, ¶132.

The Petition acknowledges that Sutardja’s degradation testing, which Petitioner relies on for limitation [1e], uses only *test* data to determine the degradation value. Petition, 55 (“Fig. 7D and paragraph [151] explain the *test*...resulting in a ‘degradation value’ for that physical address.”), 55-56 (“Paragraph [0135] discloses periodically *testing* the...NVM”). If the degradation value indicates that remapping is required, the “wear leveling module” would remap only “*new writes*” (Ex. 1042, ¶139)—not the “test” data used only to calculate the degradation value. *See* Petition, 55-56 (acknowledging that “the wear leveling

process may assign *all new writes* to another one of the memories”); *see also* Ex. 2001, ¶¶133-134.

Limitation [1e]—unlike Petitioner’s cited embodiment in Sutardja—requires *remapping an entry of addresses that fails a data integrity test* to SLC. Sutardja, by contrast, performs a data integrity test on *test data*, and based on the results, at best remaps a *different entry*—not the same “entry” as required by [1e]. Sutardja’s “test” data is not itself “remap[ped]” upon failure of the data integrity test. Accordingly, the Petition fails to demonstrate that Sutardja discloses or suggests limitation [1e]. Ex. 2001, ¶135.

Additionally, Petitioner suggests in cursory manner that “it would *also* be obvious” to rely on the teachings of Gavens—which is not even a reference in Ground 3—for limitation [1e]. Petition, 56; *see also id.*, 17 (specifying that Ground 3 only relies on Sutardja). However, yet again, Petitioner has not presented any obviousness analysis. *Id.*, 56. Petitioner, for example, fails to provide any motivation to combine Sutardja and Gavens’ disparate teachings or discussion of reasonable expectation of success. This alternative mapping should also be rejected. *See In re Stepan Co.*, 868 F.3d at 1346 n.1.

4. [1f]: “c) determine which of the blocks of the plurality of the blocks in the MLC and SLC non-volatile memory modules are accessed most frequently by maintaining a count of the number of times each one of the blocks is accessed”

The Petition fails to clearly map the language of limitation [1f] to the disclosure of Sutardja. Petition, 57. Instead of providing a clear mapping to the prior art, Petitioner ambiguously cites multiple embodiments of Sutardja. *Id.* Specifically, Petitioner cites a first embodiment, including ¶¶110-111 and 121 of Sutardja, allegedly disclosing a “wear leveling module [that]...tracks the number times [sic] the write and/or erase operations are performed on each physical block....” Petition, 57. Petitioner then cites another embodiment, including Figure 7A and ¶¶146-147 of Sutardja, allegedly disclosing “receiving write frequencies for logical addresses.” *Id.* (emphasis in original); *see also* Ex. 2001, ¶¶136-138.

The Petition, however, does not make clear how these different embodiments map to limitation [1f]. For example, Petitioner does not explain whether it is proposing to combine these embodiments, or whether it is presenting these embodiments as *alternative* reads for this Ground.⁸ As a result, Petitioner leaves its mapping for this Ground unclear, and it is not the duty of PO—or the Board—to

⁸ The Petition in no way indicates that it is relying on an obviousness analysis for limitation [1f].

infer or piece together Petitioner's case. Accordingly, Petitioner's ambiguous and non-committal Ground should be denied. *See GN Resound*, IPR2015-00103, Paper 13, 6; *A.C. Dispensing*, IPR2014-00511, Paper 6 at 5-6.

Notwithstanding, Petitioner's Ground also fails when "blocks" is properly construed to require "physical" blocks. *Supra* Section V.A. Petitioner cites Sutardja's embodiment in Fig. 7A and ¶¶146-147, and admits that, with this embodiment, Sutardja's controller determines the "write frequencies for *logical* addresses." Petition, 57. Limitation [1f], however, requires that the controller determine which *physical* blocks "are accessed most frequently."

5. [1g]: "d) allocate those blocks that receive the most frequent writes by transferring the respective contents of those blocks to the at least one SLC non-volatile memory module"

This Ground should be denied because Petitioner fails to clearly map Sutardja to limitation [1g]. Once again, Petitioner vaguely references multiple embodiments from Sutardja without clarifying how these embodiments map to limitation [1g]. Petition, 58 (citing Fig. 7A and ¶146, describing the "transfer to SLC...of the most frequently written LBAs"), 58-59 (citing Fig. 7C and ¶149, describing mapping based on "the number of write operations to [a] first physical block...during a predetermined time"), 59 (citing Fig. 7E and ¶153, describing mapping based on whether "the wear level of...MLC...is greater than a predetermined threshold").

The Petition does not say whether it is presenting these embodiments in the alternative or relying on some kind of combination of all three (for which Petitioner did not provide the required obviousness analysis). This Ground should be denied on this basis alone. *GN Resound*, IPR2015-00103, Paper 13, 6; *A.C. Dispensing*, IPR2014-00511, Paper 6 at 5-6.

Further, even when Petitioner's cited embodiments are considered individually, each embodiment does not disclose the requirements of limitation [1g]. Ex. 2001, ¶¶139-146.

First, regarding Petitioner's reliance on the embodiment of Fig. 7A and ¶146, this Ground fails for two reasons. To begin with, Petitioner's argument relies on an incorrect construction of "blocks" that improperly includes "logical" blocks. *See* Petition, 58 ("Paragraph [0146] discloses the controller mapping the *logical* addresses having write frequencies..."). However, "blocks" should be construed to mean only "physical" blocks. *See* Section V.A.

Additionally, Sutardja's embodiment of Fig. 7A and ¶146 does not disclose "*transferring the respective contents of those blocks* to the at least one SLC non-volatile memory module," as required by [1g]. Rather, this embodiment describes "*map[ping]*" "logical addresses where data *is to be written*" (i.e., data that has not been written yet) to a first or second memory based on the "write frequencies" for those addresses. Ex. 1042, ¶146. Thus, ¶146 relates to *directing a new write to a*

certain place (a “first NVS memory” or “second NVS memory”), as opposed to transferring contents of a block that is already in MLC/SLC to SLC, as required by [1g]. Ex. 2001, ¶¶141-142.

Second, regarding Petitioner’s reliance on the embodiment of Fig. 7C and ¶149, Petitioner again fails to explain how this embodiment results in the “*transferring the respective contents of those blocks* to the at least one SLC non-volatile memory module,” as required by [1g]. Petition, 58-59. Sutardja’s ¶149 describes a “data shift analysis” where, “if a number of write operations to a first block of the first NVS memory...is greater than...a predetermined threshold[,]” then the “control maps the logical addresses that correspond to the first block to a second block of the second NVS memory....” Ex. 1042, ¶149. Paragraph 149 does not state that the existing contents of the block is “transferred” as a result of this “map[ping],” as required by [1g]. *Id.*; Ex. 2001, ¶143. Instead, a POSA would have understood that, just like the Sutardja’s embodiment of ¶146 discussed above, this “map[ping]” of logical addresses would result in *directing a new write to a certain place*. *Id.*; *see also, e.g.*, Ex. 1042, ¶143 (“The mapping module...may map the logical addresses that are *to be written* more frequently...to the physical addresses of second solid-state nonvolatile memory.”); *contra* Ex. 1042, ¶131 (describing data that is “remapped and moved”).

In the same paragraph where Petitioner cites the embodiment of Fig. 7C and ¶149, it also ambiguously cites Sutardja's unrelated ¶¶126 and 131-132. Petition, 56. The Petition does not make clear how it is mapping these additional paragraphs to the claim language, or how their unrelated disclosures apply to that of ¶149. *Id.* Petitioner merely suggests that ¶¶126 and 131-132 disclose where data is “moved” as part of “wear leveling” and “remap[ping].” Petition, 59. But this disclosure is entirely unrelated to that of ¶149. For example, ¶126 relates to “mov[ing]” data that “is *unchanged* over a period of time,” whereas ¶149 relates to “map[ping]” data that is frequently written (i.e., frequently changed). Ex. 1042, ¶¶126, 149. And ¶¶131-132 instead disclose “remap[ping] and mov[ing]” data from a memory “*to create space*” when a memory has become “full.” Ex. 2001, ¶144.

To the extent the Petition is attempting to combine the unrelated disclosures of ¶149 with those of ¶¶126 and 131-132 (which is in no way apparent from the Petition itself), such mixing of different teachings without explanation is improper. *Abiomed*, 2017 WL 4804523, at *4 (denying institution) (“Petitioner’s challenge [incorrectly] treats the various features of [the reference’s] different embodiments as if they are interchangeable with one another.”).

Third, regarding Petitioner’s reliance on the embodiment of Fig. 7E and ¶153, this embodiment fails for similar reasons as the embodiment of ¶149 and Fig. 7C. Petition, 59. Sutardja’s ¶153 only states that the “control *maps* all the logical blocks

to physical blocks of the second NVS memory,” and says nothing about “*transferring the respective contents of...blocks*,” as required by [1g]. Petitioner also ambiguously cites ¶¶128, 164, and 168 of Sutardja, but again fails to make clear how these paragraphs relate to the disclosure of ¶153. Petition, 59 (including an “*Also*” cite to ¶¶128, 164, and 168). Even so, the Petition does not allege that ¶¶128, 164, and 168 disclose the missing limitation—“transferring the respective contents of...blocks.” *See* Ex. 2001, ¶145.

Furthermore, Petitioner’s citation to the embodiment of Fig. 7E and ¶153 also fails because this embodiment describes mapping “*all* the logical blocks” from a first memory to a second memory once the “wear level of the first NVS memory is greater than a predetermined threshold.” Ex. 1042, ¶153. Sutardja ¶153 describes mapping the first memory to the second memory when the memory *as a whole* reaches a certain “wear level.” *Id.* In contrast, limitation [1g] requires determining the “*blocks* that receive the most frequent writes,” and transferring the respective contents of those blocks. *See* Ex. 2001, ¶146.

Accordingly, even when each of Petitioner’s ambiguously cited embodiments of Sutardja is considered individually, Petitioner fails to show that any discloses the requirements of limitation [1g]. For this reason, this Ground should be denied.

VIII. CONCLUSION

For the foregoing reasons, Petitioner has not established that there is a reasonable likelihood that it will prevail on any ground with respect to even one of the Challenged Claims. Accordingly, the Petition should be denied.

Date: June 20, 2025

Respectfully submitted,

/Alan Whitehurst/

Alan Whitehurst
Reg. No. 43,263
awhitehurst@mckoolsmith.com
Arvind Jairam
Reg. No. 62,759
ajairam@mckoolsmith.com
Christopher McNett
Reg. No. 64,489
cmcnett@mckoolsmith.com
Christian Dorman
Reg. No. 79,118
cdorman@mckoolsmith.com
McKool Smith, P.C.
1717 K St. NW, Suite 1000
Washington, DC 20006
Telephone: (202) 370-8300
Facsimile: (202) 370-8344

Counsel for Patent Owner

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Patent Owner hereby certifies that the word count for the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE ON THE MERITS AND OTHER NON-DISCRETIONARY CONSIDERATIONS** is 12,370 words, which is within the limit set in 37 C.F.R. §42.24(b)(1). This word count does not include the table of contents, table of authorities, exhibit list, certificate of word count, and certificate of service. This word count was prepared using Microsoft Word.

Date: June 20, 2025

By: /Alan Whitehurst/
Alan Whitehurst, Reg. No. 43,263

CERTIFICATE OF SERVICE

I hereby certify, in accordance with 37 C.F.R. § 42.205, that the foregoing **PATENT OWNER'S PRELIMINARY RESPONSE ON THE MERITS AND OTHER NON-DISCRETIONARY CONSIDERATIONS** was served via electronic mail on June 20, 2025 on the following counsel of record for Petitioner:

Cono A. Carrano
ccarrano@akingump.com
Ryan S. Stronczer
rstronczer@akingump.com
Brock F. Wilson
bfwilson@akingump.com
Akin Gump Strauss Hauer & Feld LLP
2001 K St. NW
Washington, DC 20006
Telephone: (202) 887-4000
Facsimile: (202) 887-4288

George Andrew Lever Rosbrook
arosbrook@akingump.com
Akin Gump Strauss Hauer & Feld LLP
112 E. Pecan Street, Suite 1010
San Antonio, TX 78205
Telephone: (210) 781-7026

AG-KINGSTON-VERVAIN@akingump.com

Dated: June 20, 2025

By: /Alan Whitehurst/
Alan Whitehurst, Reg. No. 43,263