UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICRON TECHNOLOGY, INC.,
Petitioner,

v.

VERVAIN, LLC,
Patent Owner.

Case No.: IPR2021-01550
U.S. Patent No. 10,950,300
Original Issue Date: March 16, 2021

Title: LIFETIME MIXED LEVEL NON-VOLATILE MEMORY SYSTEM

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 10,950,300 PURSUANT TO 35 U.S.C. §§ 311-319 and 37 C.F.R. § 42

TABLE OF CONTENTS **Page**

I. INTRODUCTION 1
II. REQUIREMENTS FOR PETITION FOR *INTER PARTES* REVIEW 3
A. Grounds for Standing (37 C.F.R. §42.104(a)) 3
B. Notice of Lead and Backup Counsel and Service Information (37 C.F.R. §§ 42.8(b)(3-4), 42.10(a)) 3
C. Notice of Real Party-in-Interest (37 C.F.R. § 42.8(b)(1)) 4
D. Notice of Related Matters (37 C.F.R. § 42.8(b)(2)) 4
E. Fee for *Inter Partes* Review 5
F. Proof of Service 5
III. IDENTIFICATION OF CLAIMS BEING CHALLENGED (37 C.F.R. § 42.104(B)) 5
IV. THE BOARD SHOULD NOT EXERCISE ITS DISCRETION TO DENY INSTITUTION 7
A. The Parallel District Court Litigation Does Not Weigh Against Institution 7
B. Petitioner's Arguments Are Not Duplicative 12
V. THE 300 PATENT 12
A. Technological Background 12
1. Volatile, Non-volatile, and Flash Memory 12
2. Programming Flash, and SLC and MLC Flash Memory Cells 13
3. Flash Architecture 14
4. Caching 15
5. Logical Addresses, Physical Addresses, Bad Block Replacement, and Wear Leveling 16
6. Flash Translation Layer ("FTL") 17
7. Speed and Wear-Leveling Considerations for MLC and SLC Cells 18

Vervain Ex. 2016, p. 2 of 81
Vervain v. Phison
PGR2024-00047

8. Data Integrity Tests 19
B. Summary of the 300 Patent's Disclosure 19
C. The 300 Patent's Prosecution History 21
VI. CLAIM CONSTRUCTION 23
A. "data integrity test" (claims 1 and 12) 23
B. "comparing the stored data to the retained data in the random access volatile memory" (claims 1 and 12) 25
C. "periodically" 27
D. Other Terms 28
VII. HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE 28
A. Prior Art Overview 28
1. Dusija 28
2. Sutardja 30
B. Level of Ordinary Skill in the Art 31
C. Ground 1: Dusija In View Of The Knowledge Of A POSA Renders Obvious Claims 1-9 And 11-12 31
1. Claim 1 32
a. [1.PRE] "A system for storing data comprising:" 32
b. [1.A.1] "memory space containing volatile memory space and nonvolatile memory space;" 33
c. [1.A.2] "wherein the nonvolatile memory space includes both multilevel cell (MLC) memory space and single level cell (SLC) memory space;" 37
d. [1.B] "at least one controller to operate memory elements and associated memory space;" 38
e. [1.C] "at least one MLC nonvolatile memory element that can be mapped into the MLC memory space;" 40
f. [1.D] "at least one SLC nonvolatile memory element that can be mapped into the SLC memory space;" 41

g. [1.E] "at least one random access volatile memory;" 42

h. [1.F] "an FTL flash translation layer, wherein the at least one controller, or FTL, or a combination of both maintain an address table in one or more of the memory elements and random access volatile memory" 42

i. [1.G.1] "the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access volatile memory for storage of data therein" 46

j. [1.G.2] "the controller, in at least a Write access operation to the MLC nonvolatile memory element, operable to store data in the MLC nonvolatile memory element and retain such stored data in the random access volatile memory;" 47

k. [1.H] "the controller performing a data integrity test on stored data in the MLC nonvolatile memory element after at least a Write access operation performed thereon by comparing the stored data to the retained data in the random access volatile memory;" 48

l. [1.I] "wherein the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and" 50

m. [1.J] "wherein a failure of the data integrity test performed by the controller results in a remapping of the address space to a different physical range of addresses and transfer of data corresponding to the stored data to those remapped physical addresses from those determined to have failed the data integrity test to achieve enhanced endurance." 52

2. Claim 2: "The system of claim 1, wherein the FTL flash translation layer is a software module, or a firmware module containing software updates." 54

3. Claim 3: "The system of claim 1, wherein at least one of the random access volatile memory or the MLC and SLC nonvolatile memory elements are embedded in the at least one controller." 54

4. Claim 4: "The system of claim 1, wherein the MLC and SLC nonvolatile memory elements comprise flash memory." 55

5. Claim 5: "The system of claim 1, wherein the random access volatile memory is dynamic random access memory." 55

6. Claim 6: "The system of claim 1, wherein the random access volatile memory is static random access memory." 56

7. Claim 7: "The system of claim 1, wherein the controller, upon detection of a failure of the data integrity test, remaps the data to the SLC nonvolatile memory element." 57

8. Claim 8: "The system of claim 7, wherein the SLC memory element has a higher endurance than the MLC memory element." 57

9. Claim 9: "The system of claim 1, wherein the MLC is a multilevel cell, wherein the multilevel cell stores at least 2 bits per cell." 58

10. Claim 11: "The system of claim 1 wherein the MLC allows a single cell to store multiple bits." 58

11. Claim 12: 58

    a. [12.PRE] "A system for storing data comprising:" 58

    b. [12.A] "memory space containing volatile memory space and nonvolatile memory space, wherein the nonvolatile memory space includes both multilevel cell (MLC) space and single level cell (SLC) space;" 59

c. [12.B] "at least one controller to operate memory elements and associated memory space, and to maintain an address table in one or more of the memory elements;" 59

d. [12.C] "at least one MLC nonvolatile memory element that can be mapped into the nonvolatile memory space;" 59

e. [12.D] "at least one SLC nonvolatile memory element that can be mapped into the nonvolatile memory space;" 59

f. [12.E] "at least one random access volatile memory;" 60

g. [12.F] "the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access volatile memory for storage of data therein, the controller, in at least a Write access operation to the MLC nonvolatile memory element, operable to store data in the MLC nonvolatile memory element and retain such stored data in the random access volatile memory;" 60

h. [12.G] "the controller performing a data integrity test on stored data in the MLC nonvolatile memory element after at least a Write access operation performed thereon by comparing the stored data to the retained data in the random access volatile memory;" 60

i. [12.H] "wherein the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and" 60

j. [12.I] "wherein a failure of the data integrity test performed by the controller results in a remapping of the address space to a different physical range of addresses and transfer of data corresponding to the stored data to those remapped physical addresses from those determined to have failed the data integrity test to achieve enhanced endurance." 61

D. Ground 2: Dusija In View Of Sutardja And The Knowledge Of A POSA Renders Obvious Claim 10 61

1. Claim 10: "The system of claim 1, wherein the contents of frequently accessed portions of the memory space are periodically moved from the MLC space to the SLC space." 61

2. Motivation to Combine 63

VIII. CONCLUSION 65

TABLE OF AUTHORITIES

Page(s)

Cases

Apple Inc. v. Maxell, Ltd.,
IPR2020-00204, Paper 11, 15-17 (PTAB June 19, 2020).....10

Apple v. Fintiv,
IPR2020-00019, Paper 11, 11.....*passim*

Juniper Networks, Inc. v. WSOU Investments LLC,
IPR2021-00538, Paper 9, 13 (PTAB Aug. 18, 2021).....9, 11

Nvidia Corp. v. Invensas Corp.,
IPR2020-00603, Paper 11, 23 (PTAB Sept. 3, 2020).....11

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) (*en banc*).....23

Sand Revolution II, LLC v. Continental Intermodal Grp. – Trucking LLC,
IPR2019-01393, Paper 24, 11-12 (PTAB June 16, 2020).....10, 12

Vervain, LLC v. Micron Technology, Inc. et al.,
Case No. 6:21-cv-00487 (W.D. Tex., filed May 10, 2021)
.....4, 5, 7, 10

Vervain, LLC v. Western Digital Corporation,
Case No. 6:21-cv-00488 (W.D. Tex., filed May 10, 2021).....5

Statutes

35 U.S.C. § 314(a) 7

35 U.S.C. § 314(a) and § 325(d).....5

Other Authorities

37 C.F.R. § 42.10(b) 4

37 C.F.R. § 42.15(a).....5

Vervain Ex. 2016, p. 8 of 81
Vervain v. Phison
PGR2024-00047

37 C.F.R. §42.104(a).....3

37 C.F.R. § 42.104(B).....5

37 C.F.R. § 42.108(a). Section IV 5

157 Cong. Rec. S5429 (Sept. 8, 2011) (statement of Sen. Kyl).....8

LISTING OF EXHIBITS

Exhibit	Description
1001-1006	*Intentionally omitted*
1007	U.S. Patent No. 10,950,300 ("300 patent")
1008	File History of U.S. Patent No. 10,950,300
1009	Declaration of Dr. David Liu ("Liu Decl.") – IPR2021-01550
1010	U.S. Patent Application Publication No. 2011/0099460 ("Dusija")
1011	U.S. Patent Application Publication No. 2008/0140918 ("Sutardja")
1012	*Intentionally omitted*
1013	*Intentionally omitted*
1014	Betty Prince, Semiconductor Memories – A Handbook of Design, Manufacture, and Application (2d ed. 1991) ("Prince")
1015	U.S. Patent No. 8,120,960 ("Varkony")
1016	U.S. Patent No. 7,000,063 ("Friedman")
1017	U.S. Patent Application Publication No. 2005/0251617 ("Sinclair")
1018	Jan Axelson, USB Mass Storage: Designing and Programming Devices and Embedded Hosts (2006) ("Axelson")
1019	Rino Micheloni et al., Inside NAND Flash Memories (1st ed. 2010) ("Micheloni")
1020	U.S. Patent Application Publication No. 2011/0115192 ("Y. Lee")
1021	U.S. Patent No. 7,453,712 ("Kim")
1022	U.S. Patent Application Publication No. 2011/0096601 ("Gavens")
1023	U.S. Patent No. 8,078,794 ("C. Lee")

Exhibit	Description
1024	U.S. Patent No. 7,733,729 ("Boeve")
1025	Microsoft Computer Dictionary, Fifth Edition, 2002, definition of read-after-write
1026	Merriam-Webster's Collegiate Dictionary, Eleventh Edition, 2006, definition of periodic
1027	*Intentionally omitted*
1028	U.S. Patent Application Publication No. 2010/0172180 ("Paley")
1029	U.S. Patent No. 7,853,749 ("Kolokowsky")
1030	U.S. Patent Application Publication No. 2010/0017650 ("Chin")
1031	European Patent Specification No. EP 2.291.746 B1 ("Radke")
1032	*Intentionally omitted*
1033	U.S. Patent Application Publication No. 2006/0053246 ("S. Lee")
1034	Complaint for Patent Infringement, Dkt. No. 1, *Vervain, LLC v. Micron Technology, Inc., Micron Semiconductor Products, Inc., and Micron Technology Texas, LLC*, Case No. 6:21-cv-00487-ADA (May 10, 2021 W.D. Tex.)
1035	Agreed Scheduling Order, Dkt. No. 24, dated September 16, 2021, in *Vervain, LLC v. Micron Technology, Inc., Micron Semiconductor Products, Inc., and Micron Technology Texas, LLC*, Case No. 6:21-cv-00487-ADA
1036	Vervain's Preliminary Infringement Contentions, dated August 6, 2021, in *Vervain, LLC v. Micron Technology, Inc., Micron Semiconductor Products, Inc., and Micron Technology Texas, LLC*, Case No. 6:21-cv-00487-ADA
1037	Judge Albright, Order Governing Proceedings - Patent Cases (OGP 3.4), dated June 24, 2021

Vervain Ex. 2016, p. 11 of 81
Vervain v. Phison
PGR2024-00047

Exhibit	Description
1038	Scott McKeown, "WDTX 'Implausible Schedule' & Cursory Markman Order Highlighted," Ropes & Gray, Patents Post-Grant, Inside Views & News Pertaining to the Nation's Busiest Patent Court, June 2, 2021
1039	Dani Kass, Judge Albright Now Oversees 20% of New U.S. Patent Cases, Law360, March 10, 2021
1040	Brian Dipert and Markus Levy, Designing with Flash Memory (1994) ("Dipert & Levy")
1041	U.S. Patent No. 7,366,826 ("Gorobets")
1042	U.S. Patent No. 6,901,498 ("Conley")
1043	U.S. Patent No. 8,356,152 ("You")
1044-1046	*Intentionally omitted*
1047	Ashok Sharma, Advanced Semiconductor Memories, Architectures, Designs, and Applications (2003) ("Sharma")
1048	Microsoft Computer Dictionary, Fifth Edition, 2002, definitions of static RAM and volatile memory
1049	U.S. Patent No. 5,936,971 ("Harari")
1050	PCT Publication No. WO 03/027828 ("Gorobets WO")
1051	Microsoft Computer Dictionary, Fifth Edition, 2002, definition address space
1052	U.S. Patent Application Publication No. 2009/0300269 ("Radke Appl.")
1053	U.S. Patent No. 8,250,333 ("Gorobets II")
1054	Microsoft Computer Dictionary, Fifth Edition, 2002, definition of firmware

Vervain Ex. 2016, p. 12 of 81
Vervain v. Phison
PGR2024-00047

I. INTRODUCTION

Petitioner Micron Technology, Inc. ("Micron" or "Petitioner") respectfully requests *inter partes* review of claims 1-12 (the "Challenged Claims") of U.S. Patent No. 10,950,300 (Ex. 1007, "300 patent") which, according to USPTO records, is assigned to Vervain, LLC ("Vervain" or "Patent Owner"). There is more than a reasonable likelihood that Petitioner will prevail with respect to at least one Challenged Claim.

The 300 patent relates to flash memory devices that include both multi-level cell (MLC) and single-level cell (SLC) memory spaces. Flash memory devices with both MLC and SLC were well known and understood long before the 300 patent was filed, and the 300 patent does not contend otherwise. Instead, the 300 patent purports to improve the reliability of such devices by performing a "data integrity test" on data stored in MLC and, if the test fails, transferring the data elsewhere. During prosecution, the Examiner had rejected the 300 patent's claims, which included a data integrity test, as obvious. In response, the Applicant amended the independent claims to specify ***how*** the data integrity test is performed. Specifically, the amended claims recite performing a data integrity test by: (1) retaining data to be written in volatile memory; (2) writing the data to non-volatile (*e.g.*, flash) memory (a "write access operation"); (3) after the write access operation, reading back the stored data;

and (4) comparing the retained data with the stored data. After this amendment, the 300 patent's claims were allowed.

The Applicant did not disclose to the Examiner, nor did the Examiner cite, U.S. Pat. Pub. No. 2011/0099460 to Dusija. Dusija teaches performing a data integrity test using a "post-write read." As part of the test, one copy of data is cached in volatile memory while another copy of the data is written to flash memory. Then, after the write, the written data is read back and compared to the retained data. If too many errors are detected, the data is rewritten to another memory location. Dusija discloses, or at least renders obvious in view of the knowledge of a person of ordinary skill in the art, the other well-known limitations of the independent claims.

Except for dependent claim 10, the 300 patent's dependent claims are disclosed by—or represent well-understood, obvious modifications to—Dusija. Dependent claim 10 adds a "hot blocks limitation" in which "the contents of frequently accessed portions of the memory space are periodically moved from the MLC space to the SLC space." Sutardja, which Applicant did not disclose to the Examiner and which the Examiner did not cite, squarely discloses this feature (which was well known in memory systems with MLC and SLC memory spaces).

The 300 patent's claims thus represent nothing new or non-obvious. As such, Petitioner respectfully requests that the Board institute an *inter partes* review of the Challenged Claims and hold them to be unpatentable.

II. REQUIREMENTS FOR PETITION FOR *INTER PARTES* REVIEW

A. Grounds for Standing (37 C.F.R. §42.104(a))

Petitioner certifies that the 300 patent is available for IPR and that Petitioner is not barred or estopped from requesting IPR of the Challenged Claims on the grounds identified herein.

B. Notice of Lead and Backup Counsel and Service Information (37 C.F.R. §§ 42.8(b)(3-4), 42.10(a))

Pursuant to 37 C.F.R. §§ 42.8(b)(3-4) and 42.10(a), Petitioner provides the following designation of Lead and Back-Up counsel:

Lead Counsel	Back-Up Counsel
Jeremy Jason Lang Registration No. 73,604 (jlang@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 1000 Marsh Road Menlo Park, CA 94025-1015 T: 650-614-7400; F: 650-614-7401	Jared Bobrow (jbobrow@orrick.com) *Pro Hac Vice* to be submitted Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 1000 Marsh Road Menlo Park, CA 94025-1015 T: 650-614-7400; F: 650-614-7401 Parth Sagdeo Registration No. 71,275 (psagdeo@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 222 Berkeley St. Suite 2000 Boston, MA 02116 T: 617-880-1800; F: 617-880-1801

	Christopher Childers Registration No. 75,237 (cchilders@orrick.com) Postal & Hand-Delivery Address: Orrick, Herrington & Sutcliffe LLP 1152 15th St. NW Washington, DC 20005 T: 202-339-8441; F: 202-339-8500

Petitioner consents to service by electronic mail at the following addresses: PTABDocketJ3B3@orrick.com, PTABDocketJJL2@orrick.com, PTABDocketP2S7@orrick.com, PTABDocketC4C8@orrick.com, and Micron-Vervain_OHS@orrick.com.

Pursuant to 37 C.F.R. § 42.10(b), Petitioner's Power of Attorney is attached.

C. Notice of Real Party-in-Interest (37 C.F.R. § 42.8(b)(1))

Petitioner Micron Technology, Inc.—along with its subsidiaries—is the real party-in-interest.

D. Notice of Related Matters (37 C.F.R. § 42.8(b)(2))

According to USPTO assignment records, the 300 patent is currently assigned to Vervain. Vervain has asserted the 300 patent and U.S. Patent Nos. 8,981,298, 9,196,385, and 9,997,240 in a co-pending litigation, *Vervain, LLC v. Micron Technology, Inc. et al.*, Case No. 6:21-cv-00487 (W.D. Tex., filed May 10, 2021) ("Co-Pending Litigation"). Vervain also has asserted the 300 patent and U.S. Patent Nos. 8,981,298, 9,196,385, and 9,997,240 against Western Digital Corporation,

Western Digital Technologies, Inc., and HGST, Inc. in *Vervain, LLC v. Western Digital Corporation*, Case No. 6:21-cv-00488 (W.D. Tex., filed May 10, 2021).

In addition to this Petition, Petitioner is filing petitions for *inter partes* review to challenge the three other asserted patents in the Co-Pending Litigation: Petition for *Inter Partes* Review of U.S. Patent No. 8,981,298, IPR2021-01547, Petition for *Inter Partes* Review of U.S. Patent No. 9,196,385, IPR2021-01548, and Petition for *Inter Partes* Review of U.S. Patent No. 9,997,240, IPR2021-01549.

The Director and the Board should allow this Petition under 35 U.S.C. § 314(a) and § 325(d) and/or 37 C.F.R. § 42.108(a). Section IV.

E. Fee for *Inter Partes* Review

The Director is authorized to charge the fee specified by 37 C.F.R. § 42.15(a), and any other required fees, to Deposit Account No. 15-0665.

F. Proof of Service

Proof of service of this Petition on the Patent Owner at the correspondence addresses of record for the 300 patent is attached.

III. IDENTIFICATION OF CLAIMS BEING CHALLENGED (37 C.F.R. § 42.104(B))

Petitioner requests IPR of claims 1-12.

The 300 patent was filed on June 12, 2018. The patent also makes a facial claim of priority to a July 19, 2011 Provisional Application No. 61/509,257. 300 patent, p. 2. For purposes of this petition only, it is assumed that the 300 patent's

Vervain Ex. 2016, p. 17 of 81
Vervain v. Phison
PGR2024-00047

claims are entitled to the benefit of this July 19, 2011 date.

Petitioner's grounds rely on the following references:

(1) U.S. Patent Application Publication No. 2011/0099460 (Ex. 1010, "Dusija"): Dusija was filed on December 18, 2009 and published on April 28, 2011. It is prior art to the 300 patent under at least §§ 102(a) and (e).

(2) U.S. Patent Application Publication No. 2008/0140918 (Ex. 1011, "Sutardja"): Sutardja was filed on December 7, 2007 and published on June 12, 2008. Sutardja is prior art to the 300 patent under at least §§ 102(a), (b), and (e).

Petitioner challenges the claims on the following grounds:

Ground 1: Claims 1-9 and 11-12 are obvious over Dusija in view of the knowledge of a person of ordinary skill in the art (a "POSA");

Ground 2: Claim 10 is obvious over Dusija and Sutardja in view of the knowledge of a POSA.

None of the above references are cited on the face of the 300 patent and none were considered during prosecution of the 300 patent.

These grounds are supported by the declaration of Dr. David Liu (Ex. 1009, "Liu Decl.").

Vervain Ex. 2016, p. 18 of 81
Vervain v. Phison
PGR2024-00047

IV. THE BOARD SHOULD NOT EXERCISE ITS DISCRETION TO DENY INSTITUTION

A. The Parallel District Court Litigation Does Not Weigh Against Institution

Petitioner respectfully requests that the Board not exercise its discretion to deny institution pursuant to 35 U.S.C. § 314(a). On May 10, 2021, Vervain sued Micron, Micron Semiconductor Products, Inc., and Micron Technology Texas, LLC in the Western District of Texas, asserting the 300 patent and three other patents. Ex. 1034. Micron had no pre-suit notice of the 300 patent. Nevertheless, approximately four and a half months later, Micron filed this Petition as well as petitions on the three other asserted patents. At the time of filing this Petition, no substantial litigation activity has occurred.[1] On August 6, 2021, Vervain served its preliminary infringement contentions, which identify the claims it is asserting. Given that the Co-Pending Litigation is still in its very early stages, and discovery has not commenced, Petitioner's diligence weighs heavily in favor of institution.

Should Patent Owner argue that the Board should deny institution in its discretion under the "the *Fintiv* factors," and if the Board were to entertain such an

---

[1] On July 9, 2021, Micron filed a Rule 12(b)(6) Motion to Dismiss the Complaint because the Complaint is devoid of any factual allegations that plausibly allege infringement. The Court has not ruled on this motion.

argument, Petitioner respectfully requests that it be afforded an opportunity to submit a responsive brief to Patent Owner's arguments.

In any event, the Board should not exercise its discretion to deny this Petition. First, doing so would unfairly close the Board's doors to Petitioner. Micron was extraordinarily diligent in analyzing the prior art and preparing this Petition (along with three others) to file as early as it did.

Second, the *Fintiv* factors weigh in favor of institution. Under *Fintiv* factor three (investment in the parallel proceeding), *Fintiv* notes: "[i]f the evidence shows that the petitioner filed the petition expeditiously, such as promptly after becoming aware of the claims being asserted, this fact has weighed against exercising the authority to deny institution under *NHK*." *Apple v. Fintiv*, IPR2020-00019, Paper 11, 11. Here, Petitioner filed approximately four and a half months after receipt of the complaint and approximately six weeks after infringement contentions were served (which identified the asserted claims for the first time).[2] Moreover, to date, no court resources have been devoted to analyzing prior art, invalidity, or any other substantive issue in this proceeding. No claim construction has occurred, a motion

---

[2] Denying institution would negate Congressional intent to "afford defendants a reasonable opportunity to identify and understand the patent claims that are relevant to the litigation." 157 Cong. Rec. S5429 (Sept. 8, 2011) (statement of Sen. Kyl).

to dismiss is pending, and there has been no meaningful fact or expert discovery. When the Board issues its institution decision on this Petition, fact discovery will be in its infancy. *See* Ex. 1035 (fact discovery to begin January 21, 2022 and close August 12, 2022). Further, expert discovery is not to be completed until October 7, 2022. *Id*. And any district court claim construction proceedings that occur before institution would add to the efficiency of this IPR proceeding because the parties will submit any district court claim construction materials to the Board. On facts nearly identical to these, the Board found this factor to weigh substantially against exercising discretion to deny institution because "while the scheduled date for a *Markman* hearing ha[d] passed, much of the invested effort [wa]s unconnected to the patentability challenges." *Juniper Networks, Inc. v. WSOU Investments LLC*, IPR2021-00538, Paper 9, 13 (PTAB Aug. 18, 2021) (granting institution and stating that "the substantial work that remains on invalidity issues in the parallel district court litigation and Petitioner's expeditious filing of its Petition substantially outweighs the minimal investment so far").

Under *Fintiv* factor six (other considerations), *Fintiv* notes that if the merits of the Petition are strong, which is the case here, institution of a trial may "serve the interest of overall system efficiency and integrity because it allows the proceeding to continue in the event that the parallel proceeding settles or fails to resolve the patentability question presented in the PTAB proceeding." *Apple v. Fintiv*,

IPR2020-00019, Paper 11, 15. Vervain has already brought two patent infringement lawsuits against two memory manufacturers, and others are likely in line. What's more, Micron's petition challenges more claims than Vervain is asserting in district court, so an IPR trial will resolve issues that the parallel proceeding will not reach.

The fourth *Fintiv* factor (overlap of issues between the district court and IPR) heavily favors institution and should be given significant weight. In the Co-Pending Litigation, Patent Owner does not assert claim 6, though this Petition challenges this claim. *Compare* Ex. 1036, 2, *with* Section III. Moreover, should the Board institute an IPR proceeding on the 300 patent, Micron further stipulates that it will not pursue any instituted grounds as invalidity defenses in the District Court, thus eliminating any overlap in issues. The Board has found that such stipulations weigh in favor of institution. *See Sand Revolution II, LLC v. Continental Intermodal Grp. – Trucking LLC*, IPR2019-01393, Paper 24, 11-12 (PTAB June 16, 2020) (informative); *Apple Inc. v. Maxell, Ltd*., IPR2020-00204, Paper 11, 15-17 (PTAB June 19, 2020). Accordingly, this factor weighs strongly in favor of institution.

The second *Fintiv* factor (the proximity of the trial date to the final written decision) and the fifth *Fintiv* factor (whether the parties are the same) favor institution. The parties are almost always the same, and this case is no exception. Nonetheless, the Board has noted that the fifth factor favors institution if the Board is "likely to reach the merits at or around the same time as the district court case."

Vervain Ex. 2016, p. 22 of 81
Vervain v. Phison
PGR2024-00047

Nvidia Corp. v. Invensas Corp., IPR2020-00603, Paper 11, 23 (PTAB Sept. 3, 2020). Here, the Court typically does not set a trial date until after the *Markman* hearing. *See* Ex. 1037, 11. However, it recently set a "first trial" date (the scheduling order applies to Micron and WD) of 1/23/2023, "subject to the Court's availability." Ex. 1035, 1, n. 1, 3. Therefore, it remains uncertain when exactly Micron's trial will proceed. And while this date falls a few months before the projected final written decision date, the judge presiding over the district court case, Judge Alan Albright, has an exploding docket that has made it difficult for the Court to hold trials on the dates they are scheduled. *See* Exs. 1038, 1039. Thus, even if the final written decision date falls after the "scheduled trial date," the Board is still likely to reach the merits of the Petition before or around the same time as the district court actually holds a trial. In a recent Board institution decision, where a parallel trial in the Western District of Texas was (at best) estimated to begin "a month" before a Final Written Decision, the Board noted that "[o]n this record, it is unclear whether the trial date in the parallel district court litigation with respect to the … patent will be set to proceed before the due date for a final written decision." *Juniper Networks*, IPR2021-00538, Paper 9, 9-10. For the same reasons, the Board should institute in this proceeding.

Finally, the first *Fintiv* factor (potential for a district court stay) is neutral. Neither party has yet requested a stay, so at worst this factor is neutral because the

Board "will not attempt to predict" how the district court will proceed. *Sand Revolution II*, IPR2019-01393, Paper 24, 7.

On balance, the *Fintiv* factors favor institution.

B. Petitioner's Arguments Are Not Duplicative

The arguments and prior art relied on in this Petition are entirely different from those addressed by the Patent Office during prosecution. None of the prior art references discussed in this Petition were cited or considered. The claims were allowed because the Applicant amended the claims to recite performing a data integrity test by: (1) retaining data to be written in volatile memory; (2) writing the data to non-volatile (*e.g.*, flash) memory (a "write access operation"); (3) after the write access operation, reading back the stored data; and (4) comparing the retained data with the stored data. *See* Section V.C. This Petition presents a new and non-duplicative issue because the Dusija reference discloses a data integrity test performed in this manner.

V. THE 300 PATENT

A. Technological Background

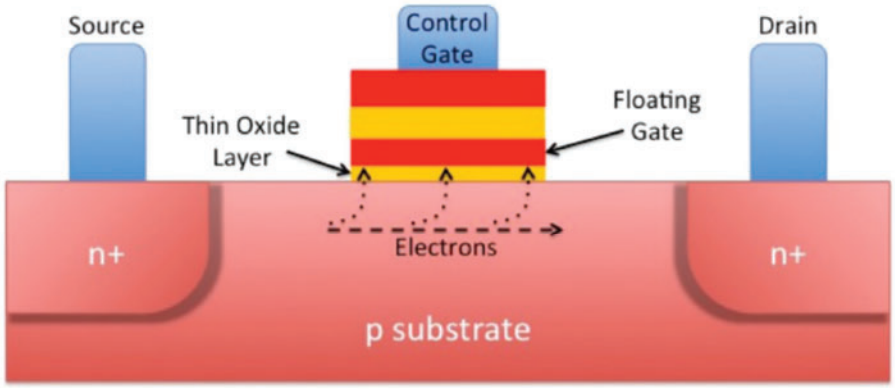
1. Volatile, Non-volatile, and Flash Memory

Computer system memory may either be "volatile" or "non-volatile." ***Volatile*** memory, such as dynamic random access memory (DRAM) and static random-access memory (SRAM), needs continuous power to preserve stored data. ***Non-volatile*** memory (sometimes called NVM or NV memory), such as flash memory

and disk drive storage, maintains its contents when power is turned off. Volatile memory tends to be faster than non-volatile memory but, of course, loses all of its data when a computer is turned off. Liu Decl., ¶¶ 42-43.

Flash memory has been a known memory type for several decades. Flash has the advantage of being smaller in size relative to traditional non-volatile storage, *e.g.*, disk drives. Liu Decl., ¶ 44.

The memory cell of flash memory can be a floating gate transistor:



http://flashdba.com/2015/01/09/understanding-flash-floating-gates-and-wear/

See also, Ex. 1014, Prince, 54-55; Liu Decl., ¶ 45.

2. Programming Flash, and SLC and MLC Flash Memory Cells

The amount of charge stored on the floating gate of a flash memory cell indicates the value that the cell stores. A typical way to program (*i.e.*, write to) a flash cell is to inject charge onto the floating gate, thereby setting the value that the cell stores. Liu Decl., ¶¶ 47-48.

When the amount of stored charge in a flash memory cell can correspond to one bit only, *i.e.*, 0 or 1, such cells are referred to as single-level cells ("SLC"). Ex. 1015, "Varkony," 3:4-7, 7:24-35. When the amount of stored charge in a flash memory cell can correspond to more than one bit of information, (more than two different values), *e.g.*, 00, 01, 10, or 11, such cells are referred to as multi-level cells ("MLC"). *Id.*; Liu Decl., ¶ 50.

SLC and MLC have advantages and disadvantages relative to the other. All flash memory cells, however, "wear out," *i.e.*, after a certain number of erase and program (write) operations, they become defective. *E.g.*, Ex. 1016, "Friedman," 1:19-37. But SLC can sustain many more program operations than MLC before becoming defective. SLC also can be programmed more quickly than MLC. On the other hand, MLC can store more data per unit cost than SLC (and stores data more densely). Liu Decl., ¶ 51.

3. Flash Architecture

Unlike most memories, an area of memory in flash memories must be empty before it can be written to. Thus, flash memories separately first "erase" and then "program" (*i.e.*, write to) memory locations. For efficiency, both erase and program operations are performed on ranges of memory addresses rather than any single address. The range of addresses for a single program is known as a "page," and the

range of addresses for a single erase is known as a "block." Blocks are typically larger than pages. Liu Decl., ¶¶ 52-56.

Erasing a block takes a relatively long time, *e.g.*, an order of magnitude longer than a read or program operation. Ex. 1017, "Sinclair," [0008]. Thus, it was known to a POSA that it was undesirable to write a page to a location that was not already erased. *Id.*, [0009]. Also, in typical computer system programs using flash memory, certain files, and thus certain pages in flash memory, are modified much more often than other pages. Because flash memory cells wear out, this makes pages wear out at an uneven rate. In view of the above, well-known wear-leveling techniques with corresponding logical-to-physical addressing were introduced to ensure that (1) previously erased blocks would be immediately available to program, and (2) the program operations would be spread evenly among the device pages so the pages would evenly wear. Liu Decl., ¶¶ 57-59.

4. Caching

Computer systems have for decades used caches. A cache, put simply, is a storage medium used in conjunction with another, slower storage medium. For example, SRAM placed inside a processor has long been used as a cache for a computer's main memory, which is typically DRAM. Ex. 1047, "Sharma," 21-24. In a similar manner, a DRAM cache is used to store data read from hard disk drives. Ex. 1019, "Micheloni," 14; Liu Decl., ¶ 60.

It was well-known to use a variety of caches in flash memory systems. For example, through the well-known process of write buffering, a volatile memory cache—typically composed of DRAM or SRAM—is used to store data to be written to the flash device. Micheloni, 132 (SRAM cache), 44 (DRAM cache). One advantage of using a write cache is that a number of small data writes can be consolidated into a single page, thus reducing the number of writes to flash memory. *Id.*, 528-29. Another known benefit is that the volatile memory cache can be written to more quickly. *Id.*, 44. A further known benefit is that the volatile memory cache has high write endurance, and so can be written to repeatedly without degrading in the manner that a flash cell would. *Id.* Finally, another known benefit is that the data in the cache can be offloaded to flash when the device is idle. *Id.*

As the above illustrates, various memory types were used as caches, including, for example, SRAM and DRAM. Micheloni, 132 (SRAM cache), 44 (DRAM cache); Liu Decl., ¶¶ 61-62.

5. Logical Addresses, Physical Addresses, Bad Block Replacement, and Wear Leveling

The logical address space is the set of addresses that the higher-level system, *e.g.*, a personal computer system (host), uses to access a flash memory. A logical address is translated ("address mapped") into a physical address before access to the actual memory, *e.g.*, flash memory, occurs. The flash device uses physical addresses, *i.e.*, the physical addresses that specifically identify physical blocks,

pages, and/or sectors in the flash memory. For example, the host directs read and write operations to logical addresses, and an address translation scheme translates the logical addresses into physical addresses. Specifically, a table (an "address map") maintained by, *e.g.*, the flash memory's controller, is used to translate logical addresses to physical addresses. Ex. 1018, "Axelson," 29-30. As an example, when the host directs a read operation to logical block 83, the flash device (or its controller) might translate this logical address to physical block 710. It was well known in the prior art for logical addresses and corresponding physical addresses to use blocks, pages, or other units.

Logical-to-physical address mapping has several well-recognized benefits, *e.g.*, allowing the flash memory to replace defective blocks ("bad blocks"). Micheloni, 42. In addition, through well-known "wear leveling," logical-to-physical block mappings can be changed in order to spread program and erase operations (both of which cause memory cells to wear out) evenly across the flash device's physical blocks. Liu Decl., ¶¶ 63-78.

6. Flash Translation Layer ("FTL")

The term "flash translation layer" or FTL describes the functionality that receives host read and write requests (which use logical addresses) and translates these requests to actual, physical reads and writes to flash chips (*i.e.*, to physical

Vervain Ex. 2016, p. 29 of 81
Vervain v. Phison
PGR2024-00047

locations). At its most basic, this involves translating logical to physical addresses. Liu Decl., ¶ 79.

An FTL is related to, but a distinct concept from, a controller. The FTL is a function, and the controller is the structure that often implements that function. Micheloni, 492 ("Usually it is the Controller . . . taking care of the FTL."). Liu Decl., ¶ 80. Thus, when the controller is performing these functions, a POSA would have understood the controller's FTL is performing these functions. Micheloni, 492, 517; Liu Decl., ¶¶ 81-82, 84.

The 300 patent's background section uses the term "FTL" in a manner consistent with the above. 300 patent, 3:3-7, 3:11-16; Liu Decl., ¶ 83.

7. Speed and Wear-Leveling Considerations for MLC and SLC Cells

The write speed for MLC is markedly slower than for SLC. Ex. 1020, Y. Lee, [0005]. Thus, it was well known that the system will operate faster if frequently updated data is written to SLC instead of MLC. *Id*. It was commonly known that implementing a combined system that includes both MLC and SLC memory cells provided the benefits of large storage capacity due to MLC and high-speed data access due to SLC. *Id*. Conventional MLC can afford roughly 10,000 erase operations in its operative lifespan while SLC affords between 100,000 – 1,000,000 erase operations during its lifetime. *Id*., [0006]. In order to maximize the lifetime of the devices, it was known to write frequently updated blocks ("hot blocks") to

SLC due to its higher erase cycle capacity and faster speed. *Id*., [0005]; Liu Decl., ¶¶ 85-86.

8. Data Integrity Tests

Data integrity tests, broadly speaking, involve writing data to memory and, after the write operation has been completed, performing an operation (usually a read operation) to ensure that the data was written without errors. *See* Ex. 1022, "Gavens," [0012]. Performing a data integrity test on flash memory was well-known in the art, *id*., and the term "data integrity test" was in use well before the 300 patent was filed. Ex. 1024, "Boeve," 10:41-53; Ex. 1025, "Microsoft Dictionary," 440 ("read-after-write" is a "feature of certain data storage devices…in which the device reads data immediately after it is written as a means of verifying data integrity."); Liu Decl., ¶ 87.

B. Summary of the 300 Patent's Disclosure

The specification of the 300 patent purports to teach two techniques for improving the reliability of flash devices that have at least one MLC memory module and at least one SLC memory module. *See* 300 patent, 4:17-51; Liu Decl., ¶ 88.

In the first technique, the system performs a "data integrity test" on data stored ***in an MLC memory space*** and, if the test fails, the system moves the data ***to an SLC memory space***. 300 patent, 5:61-67. More specifically, the 300 patent explains that a "data integrity test" is conducted after each write operation to memory to ensure

that the data was written correctly. *Id.*, 4:30-32. The data to be written to NAND flash is initially stored in a bank of DRAM. *Id.*, 5:59-61, 6:17-22. The device controller then reads the data and stores the data into its memory within the device controller. *Id.*, 6:20-22. While maintaining a copy of the data in device controller memory (which it refers to as the "retained data"), the controller writes the data to an MLC physical address range (*e.g.*, a block) of NAND flash corresponding to the data's logical address range. *Id.*, 6:25-30. After the write operation, the controller reads the contents of the physical address range back into a separate portion of device controller memory (which it refers to as the "newly stored data"). *Id.*, 6:30-32. The controller compares the newly stored data to the maintained (or retained) data. *Id.*, 6:33-38. If the newly stored data matches the retained data, the data integrity test succeeds. Otherwise, the data integrity test fails. If it fails, the data is rewritten to a new physical SLC address and the controller remaps the data's logical address range to a new physical address in SLC memory. *Id.*, 6:40-52; Liu Decl., ¶ 89.

The independent claims recite a "data integrity test" that includes the above process, namely, (1) retaining data to be written in volatile memory; (2) writing the data to non-volatile (*e.g.*, flash) memory (a "write access operation"); (3) after the write access operation, reading back the stored data; and (4) comparing the retained data with the stored data. Dusjia squarely discloses this type of data integrity test. *See* limitation [1.H]; Liu Decl., ¶ 90.

In the second technique, the 300 patent discloses allocating MLC blocks that are "hot blocks" (*i.e.*, those that receive frequent writes) into the SLC memory space. 300 patent, 6:53-59, 6:59-65; Liu Decl., ¶ 91.

Only dependent claim 10 relates to this hot blocks limitation. It was well known, however, to transfer the data of frequently written MLC blocks to SLC blocks, as Sutardja discloses. *See* Section VII.D.1 (Sutardja).

C. The 300 Patent's Prosecution History

As initially filed, claim 1 of the 300 patent did not recite a data integrity test. The Examiner rejected the claim as obvious in view of U.S. Patent Application Publication No. 2010/0172179 and a journal article (Goodson). Ex. 1008, 46-47. In response, the Applicant added a limitation that recited performing a data integrity test on data and, if the test failed, remapping the data to a different physical range of addresses. *Id*., 55, 59. The Examiner again rejected the claims as obvious. *Id.*, 73. The Applicant then added an additional limitation reciting that a data integrity test results in data being transferred. *Id*., 91-94. The Examiner continued to find the claims to be obvious. *Id.*, 145-47.

The Applicant then amended the claims to recite a specific way to perform the data integrity test. First, when the controller performs a "Write access operation," it both "store[s] data in the MLC nonvolatile memory element" and "retain[s] [the] stored data in [a] random access volatile memory." Ex. 1008, 173. The controller

Vervain Ex. 2016, p. 33 of 81
Vervain v. Phison
PGR2024-00047

then reads back the data stored in MLC memory (*i.e.*, the "stored data"). *Id.*, 180. The controller performs the data integrity test "by comparing the stored data to the retained data in the random access volatile memory." *Id.*, 173. The Examiner allowed the application after this amendment. *Id.,* 173, 180, 204.

In issuing the notice of allowance, the Examiner pointed to two pages of the Applicant's response to the prior Office Action. Ex. 1008, 210 (citing *id.*, 179-180). In that response, the Applicant distinguished the claims from the prior art in two ways. First, Applicant argued that the "conventional cache" present in the prior art allegedly was not big enough to perform the retaining and comparison limitations:

> This operation [*i.e.*, a data integrity test] is not some operation that can be carried out by a conventional cache, as that cache is typically only designed to handle one block of data. In the nonvolatile memory, in order to facilitate a compare operation, if an entire block is being written to with new data, that entire block has to be retained in one portion of the memory and then written to the memory, followed by a subsequent Read of the memory, requiring that block of data to be placed in an additional area of the nonvolatile [sic] memory, thus requiring at least two blocks of memory space in the nonvolatile [sic] memory.

Id., 180.[3] In other words, Applicant distinguished prior art caches on the ground that the prior art caches allegedly were not big enough to store ***both*** (1) the retained written data (*i.e.*, a copy of the original data that was written) and (2) the read back data (*i.e.*, the read version of the written data).

Second, and relatedly, Applicant argued that because the prior art allegedly does not disclose "retain[ing] the data to be written for later comparison," the prior art does not disclose the "transfer of data corresponding to the stored data" (*i.e.*, the retained data) and a data integrity test using a "comparison" with retained data. Ex. 1008, 180.

VI. CLAIM CONSTRUCTION

The Board applies the same *Phillips* claim construction standard used by the district courts. *Phillips v. AWH Corp.,* 415 F.3d 1303 (Fed. Cir. 2005) (*en banc*).

A. "data integrity test" (claims 1 and 12)

Claims 1 and 12 recite "the controller performing a data integrity test on stored

---

[3] Applicant references above to "nonvolatile memory" are presumably typographical errors and should be "volatile memory" instead, because the 300 patent only refers to performing such actions with volatile memory (Section V.B) and Applicant clearly misstates that the limitation requires "retain[ing]" the stored data in nonvolatile memory (Ex. 1008, 179). Liu Decl., ¶¶ 96-98.

data in the MLC nonvolatile memory element after at least a Write access operation performed thereon by comparing ….” In the 300 patent claims, unlike in earlier family members of the 300 patent, the claim language itself defines the data integrity test, namely, that it occurs “after at least a Write access operation” and “by comparing ….” Accordingly, in the context of the 300 patent, Petitioner does not believe that it is necessary to construe this term. To the extent that the Board determines that a construction is necessary, in the context of the 300 patent, a POSA would have understood “data integrity test” to mean “a test conducted after a write to flash to ensure that the data was written correctly.”[4] Liu Decl., ¶¶ 105-07.

In the Summary of the Disclosure section, the 300 patent’s specification states that “[a]fter each write to (flash) memory, the controller conducts a data integrity check to ensure that the data was written correctly.” 300 patent, 4:30-32; *see also id.*, 5:61-64 (“[a]fter each write . . . the device controller 14 will . . . perform a read on the address range to ensure the integrity of the written data”).

Note that while the specification appears to suggest that the data integrity test occurs after each write, this just describes, in particular embodiments, how often the test occurs. Moreover, the claims recites that the controller must perform the data

---

[4] Petitioner advances this construction in petitions challenging earlier family members of the 300 patent. *See* Section II.D.

integrity test "after at least a Write access operation," not after all write access operations. 300 patent, 7:61-65, 8:59-63.

B. "comparing the stored data to the retained data in the random access volatile memory" (claims 1 and 12)

Claims 1 and 12 recite a "Write access operation . . . store[s] data in the MLC nonvolatile memory element and retain[s] such stored data in the random access volatile memory." "[A]fter at least a Write access operation," "the controller . . . perform[s] a data integrity test on stored data in the MLC nonvolatile memory element . . . by ***comparing the stored data to the retained data in the random access volatile memory***." (emphasis added).

In the context of the 300 patent, a POSA would have understood "comparing the stored data to the retained data" to mean "comparing a copy of the data obtained by reading the nonvolatile memory space (the stored data) to a copy of the data retained as part of a Write access operation (the retained data), wherein both copies are placed into the same volatile memory." Liu Decl., ¶¶ 108-14.

This understanding of the claim language is compelled by the Applicant's statements during prosecution in which Applicant made clear that (1) the stored data must be a read-back copy of the written flash data, (2) the retained data must be a copy of the data written to flash memory as part of a Write access operation and retained in volatile memory as part of that operation, (3) neither the stored data nor the retained data is ECC (error correction code) bits, and (4) the volatile memory is

big enough to store both copies.

During prosecution, the claims—which at the time recited a data integrity test without specifying *how* it was performed—were rejected as obvious. Ex. 1008, 145. In response, the Applicant amended the claims to recite that in a "Write access operation to the MLC nonvolatile memory element," the controller "store[s] data in the MLC nonvolatile memory element and retain[s] such stored data in the random access volatile memory." *Id.*, 173. The Applicant further amended the claimed "data integrity test" to be performed "after . . . at least a Write access operation . . . by ***comparing the stored data to the retained data in the random access volatile memory***." *Id.* (emphasis added).

In distinguishing the amended claims from the cited art, the Applicant stated: "This [data integrity test] is not some operation that can be carried out by a conventional cache, as that cache is typically only designed to handle one block of data. In the nonvolatile memory, in order to facilitate a compare operation, if an entire block is being written to with new data, that entire block has to be retained in one portion of the memory and then written to the memory, followed by a subsequent Read of the memory, requiring that block of data to be placed in an additional area of the nonvolatile [sic] memory, thus requiring at least two blocks of memory space in the nonvolatile [sic] memory." Ex. 1008, 180; *see id.*, 173 (at the same time as above quote, amending claims to recite "comparing the stored data to the retained

data ***in the random access volatile memory***" (emphasis added)); 300 patent, 6:15-38 (describing the comparison as occurring in "memory within the device controller"). In other words, the stored and retained data are copies of the originally written data to the flash memory. Indeed, the Applicant repeatedly distinguished the cited prior art on this basis, explaining that "[s]ince neither [of the cited references] retain the data to be written for later comparison, neither suggests nor discloses such an operation." Ex. 1008, 103, 180, 202.

Further, Applicant distinguished the claims from a system based on error-correcting codes ("ECC"), since an ECC-based data integrity test would not retain a copy of data in volatile memory after writing the data. Ex. 1008, 189. The Examiner found the Applicant's distinctions persuasive in allowing the claims. *Id.*, 189, 210.

Note that the specification only describes a comparison with stored and retained data that are actual copies of the original written data. 300 patent, 6:17-52.

C. "periodically"

Claim 10 recites "[t]he system of claim 1, wherein the contents of frequently accessed portions of the memory space are periodically moved from the MLC space to the SLC space."

The 300 patent does not disclaim or limit the term "periodically." Instead, it gives an example of transfers occurring "on a periodic basis" as those occurring "every 1000 writes, or every 10,000 writes." 300 patent, 6:59-65. Thus,

"periodically" should have its plain and ordinary meaning, which includes events that "occur[] repeatedly from time to time" and those that are "consisting of or containing a series of repeated stages, processes or digits." Ex. 1026, Merriam-Webster's Collegiate Dictionary, 921; Liu Decl., ¶¶ 115-16.

D. Other Terms

Petitioner believes that the other terms of the 300 patent's claims do not require further construction and can be afforded their plain and ordinary meaning.

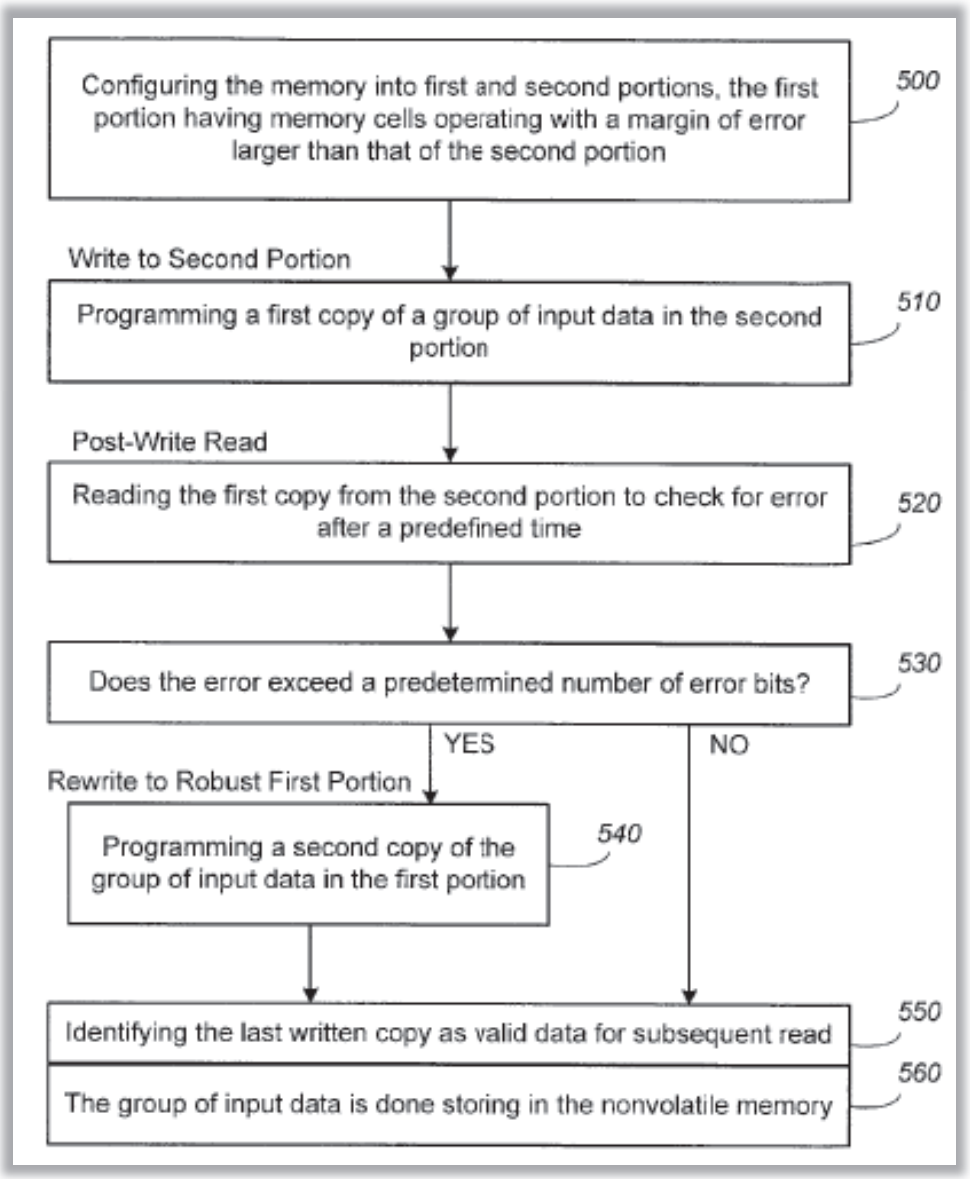
VII. HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Prior Art Overview

1. Dusija

Dusija relates to a flash memory device with a "first portion operating with less error but of lower density storage" (*i.e.*, SLC) and a "second portion operating with a higher density but less robust storage" (*i.e.*, MLC) in a memory array. Dusija, Abstract, [0019].

Just like the 300 patent, Dusija discloses a data integrity test that rewrites data if an original write to MLC fails. Specifically, it discloses a "post write read" method of checking errors, performed after data is written into the memory array. Dusija, [0111], [0121], Figures 14-15. The "post write read" process is depicted in Figure 15 below:



Id., Figure 15. In this method, data is written in the second portion of memory (*i.e.*, MLC) (step 510). *Id.* Then, after the write ("post-write"), the stored version of the data is read back in a "post write read" to determine if there are any errors (step 520). *Id.*, [0112]. The stored version of the data is compared to an "original copy," which can be cached. *Id.* As discussed below, based on Dusija's full incorporation of Paley, a POSA would have understood that Dusija's cache could be a volatile

random access memory. *Id.*, [0061]. The device determines if the number of error bits from the comparison exceeds a predetermined amount (step 530) and, if so, as depicted in Figure 14B below, rewrites the data to the first portion of memory (*i.e.*, SLC) (step 540). *Id.*, [0113], [0116]. After writing to the first portion is complete, the copy in MLC becomes obsolete, and the "block management system …. [is] updated" (which includes updating the logical-to-physical address table, *i.e.*, remapping the logical address to the new physical address in SLC). *Id.*, [0117]. As a result, subsequent host accesses will be directed to the copy in SLC (step 550). *Id.*

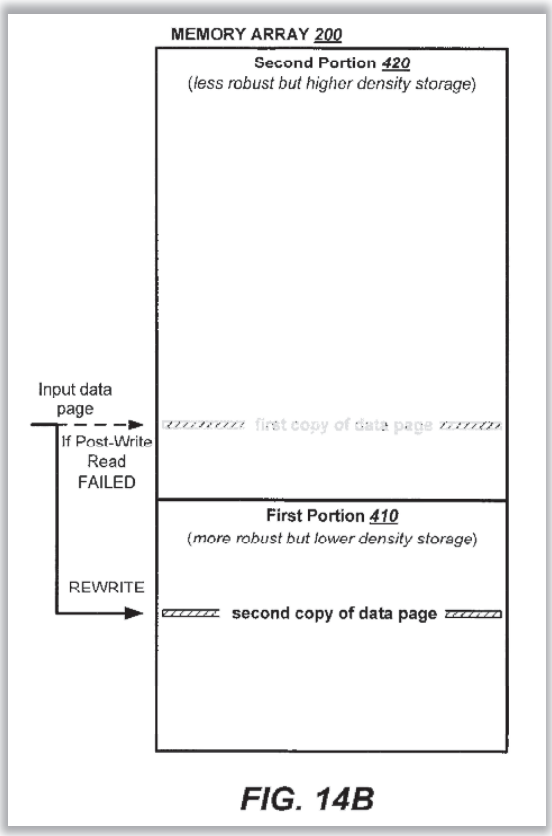


FIG. 14B

Dusija, Figure 14B; Liu Decl., ¶ 119.

2. Sutardja

Sutardja relates to "a hybrid non-volatile solid-state memories" (Sutardja, [0002]) that have a "first solid-state nonvolatile memory" (which may be MLC) and "second solid-state nonvolatile memory" (which may be SLC). Sutardja, [0108].

Just like the 300 patent's dependent claim 10, Sutardja discloses a hot blocks feature that moves data from MLC to SLC memory. Specifically, Sutardja discloses performing a "data shift analysis" whereby "if a number of write operations to a first block of the first NVS memory [*i.e.*, MLC] during a predetermined time is greater than or equal to a predetermined threshold," the system "maps the logical addresses that correspond to the first block to a second block of the second NVS memory [*i.e.*, SLC]." Sutardja, [0149], Claim 37 (first memory is MLC and second memory is SLC); Liu Decl., ¶¶ 121-122.

B. Level of Ordinary Skill in the Art

A POSA in the technology field of the 300 patent would be a person with at least a Bachelor of Science degree in electrical engineering, computer engineering, or a closely related field, along with at least 3-5 years of experience in the design of non-volatile memory devices. An individual with an advanced degree in a relevant field would require less experience in the design of non-volatile memory devices. Liu Decl., ¶ 37.

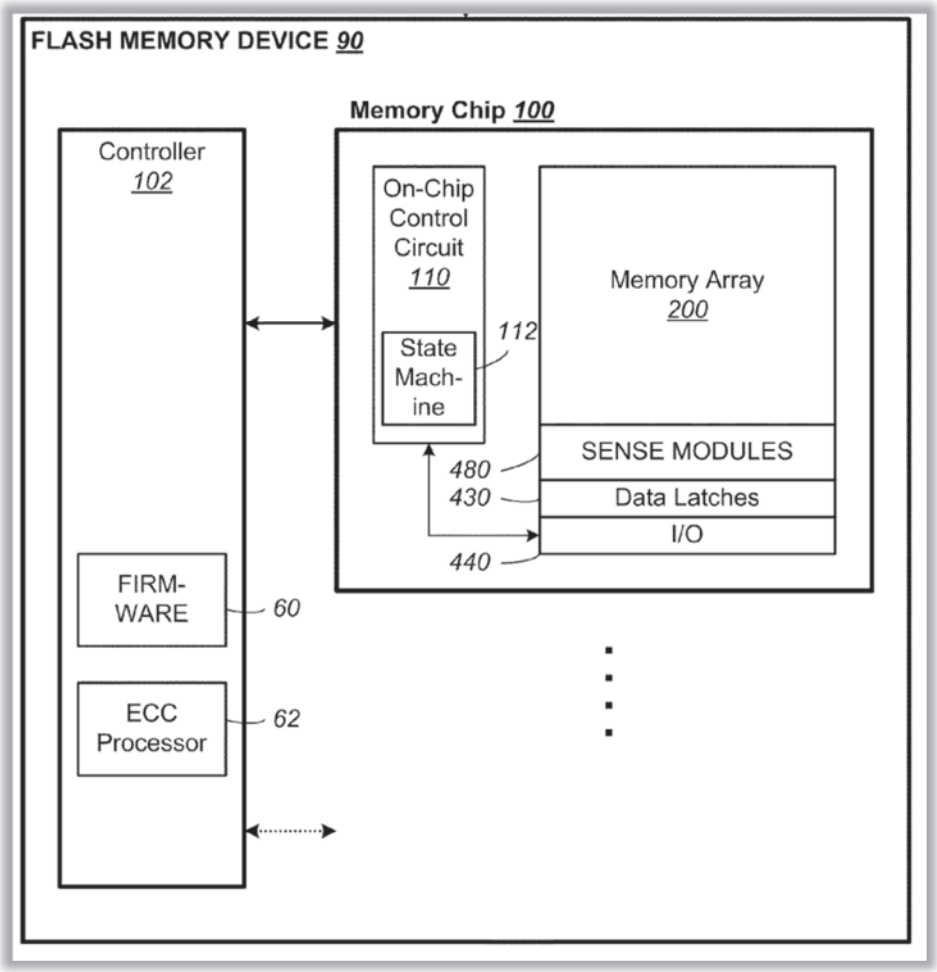
C. Ground 1: Dusija In View Of The Knowledge Of A POSA Renders Obvious Claims 1-9 And 11-12

Dusija discloses, or, at a minimum, renders obvious, claims 1-9 and 11-12.

1. Claim 1

a. [1.PRE] "A system for storing data comprising:"

To the extent the preamble is limiting, Dusija discloses "*a system for storing data*,"[5] namely, "flash memory device 90" stores data in memory array 200:



Dusija, Figure 1, [0059]; Liu Decl., ¶ 125.

[5] For the Board's convenience, the claim language in the claim mapping portions is in italics and red.

b. [1.A.1] "memory space containing volatile memory space and nonvolatile memory space;"

Dusija discloses this limitation.

Dusija discloses "*memory space containing . . . nonvolatile memory space*," namely, it discloses memory array 200 that includes flash memory cells. Dusija, [0059], [0068]; *see also* limitation [1.A.2].

Dusija discloses, or at least renders obvious, "*memory space containing volatile memory space*." Liu Decl., ¶¶ 127-28.

First, Dusija discloses a "post-write read" feature in which a comparison is made between an "original copy [of data] which may be ***cached***" and a read-back version of the data. Dusija, [0111]–[0117] (emphasis added). A POSA would have understood (and certainly would have found it obvious) that a cache is typically implemented in random access volatile memory.[6] For example, static RAM ("SRAM") is "usually reserved for use in caches" and is a type of volatile memory. Ex. 1048, Microsoft Dictionary, 497, 558. Further, during the prosecution of the

---

[6] Dusija states that its "cache" could be, "[i]n an alternative embodiment," flash memory (what Dusija calls the "first portion). Dusija, [0020]. This only reinforces that the primary embodiments (not "alternative") would be understood to employ a RAM cache and certainly does not act to limit the claim scope. Liu Decl., n. 6.

300 patent, the Examiner repeatedly took official notice that SRAM and DRAM are "volatile memory types" that are often used as "cache memory." Ex. 1008, 48, 78. The Applicant did not contest this use of official notice. Liu Decl., ¶ 129.

Second, Dusija discloses controller 102, which includes firmware 60, that manages memory operations. Dusija, Fig. 1, [0062]. A POSA would have understood, and at least found it obvious, that the controller includes RAM to buffer data and commands, execute the firmware, and manage the memory operations. Liu Decl., ¶ 130 (citing U.S. Patent Application Publication No. 2009/0327591 and Ex. 1019, Micheloni, 132 for examples of a POSA's knowledge regarding controller's RAM).

Third, Dusija fully incorporates the disclosures of Paley. Dusija, [0061]. Paley discloses a controller for a flash memory device that includes RAM ("*volatile memory space*"). Paley, Fig. 1, [0119]. Paley further states that "[t]ypically high-speed volatile RAM are employed as cache in a . . . controller cache 102 of the memory controller." *Id*., [0197]. Thus, Paley discloses this limitation and confirms that it was customary for RAM to be employed as cache in a flash memory device's controller. *Id*.; Liu Decl., ¶¶ 131-33.

Fourth, Paley fully incorporates the disclosure of U.S. Patent No. 5,936,971 to Harari et al. (Ex. 1049, "Harari"), explaining that Harari discloses "[u]sing RAM in a write cache operating with a flash memory." Paley, [0027]. Harari also discloses

this limitation and confirms it was customary, disclosing "a cache system 701 as part of the controller" for temporarily holding files prior to writing them to flash. Harari, 12:20-35 ("cache" typically RAM), 13:24-42, 13:58-14:11. Liu Decl., ¶¶ 134-36.

To the extent that Patent Owner argues that the 300 patent's volatile memory space does not include "conventional cache[s]" and that Dusija is a "conventional cache," Patent Owner is incorrect, because Dusija's cache is not a "conventional cache." During prosecution, the Applicant attempted to distinguish the claimed "volatile random access memory" from a "conventional cache." Ex. 1008, 180. In particular, Applicant distinguished prior art caches on the ground that they allegedly were not big enough to store ***both*** (1) the retained written data (*i.e.*, a copy of the original data that was written) and (2) the read back data (*i.e.*, the read version of the written data). *See* Section V.C. As Section VIII.C.1.k, limitation [1.H], illustrates, Dusija's cache is not a "conventional cache" in that it can store ***both*** the retained written data and the read back data. Moreover, it was well known for caches to be large enough to do so, *e.g.*, Harari's cache is capable of storing multiple files. Harari, 13:29-30; Liu Decl. ¶ 137.

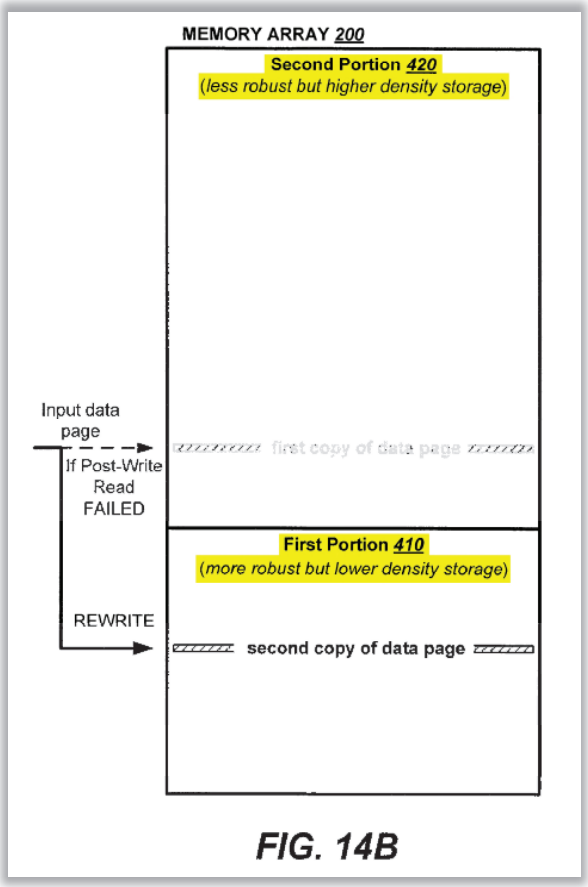
To the extent Patent Owner might argue that Dusija does not fully incorporate Paley and Harari (or the relevant portions of these references), Patent Owner is incorrect. Dusija cites Paley, stating: "A preferred block management system is disclosed in U.S. application Ser. No. 12/348,891, filed on 1 Jan. 2009, ***the entire***

disclosure of which is incorporated herein by reference." Dusija, [0061]. Moreover, Dusija specifically refers to Paley's "block management system," and Paley states that "[b]oth the cache portion and the main memory portion operate ***under a block management system*** for which cache portion is optimized." Paley, [0032]. Similarly, Paley incorporates Harari, referencing Harari's disclosure of "[u]sing RAM in a write cache operating with a flash memory." *Id.*, [0027].

At the very least, it would have been obvious to employ Dusija's controller with RAM and implement Dusija's cache as RAM in view of Paley, Harari, and/or the knowledge of a POSA. First, with respect to the hardware specifics, Dusija generically discloses a "cache." Thus, a POSA would have been motivated to consider such teachings that Dusija references (Paley and Harari) and the ordinary knowledge of a POSA. Second, using a RAM as a cache was well known and typical. *See* Section V.A.4. Third, Dusija discloses that its controller and cache must store an original copy and a read-back copy and must be suitable for supporting a comparison operation. A POSA would have immediately thought of a controller's RAM as optimal for such purposes given (1) its relatively high speed and random access capability and (2) it was routine to implement and highly predictable. *See* Section V.A.4.; Liu Decl., ¶ 138.

c. [1.A.2] "wherein the nonvolatile memory space includes both multilevel cell (MLC) memory space and single level cell (SLC) memory space;"

Dusija discloses the limitation, namely, a "memory array 200" comprising a "second portion 420" and "first portion 410":



Dusija, Figure 14B (annotated); Liu Decl. ¶¶ 139-142.

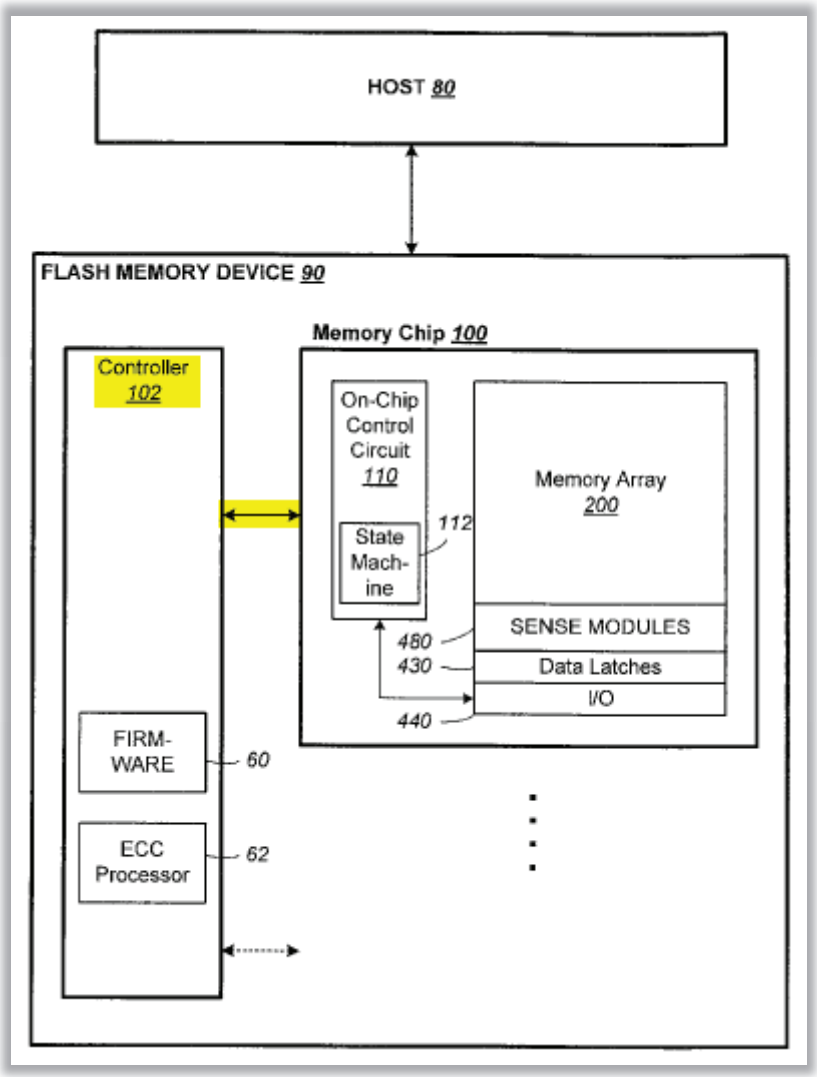
Dusija's "second portion" is a "*multilevel cell (MLC) memory space*": "the second portion 420 has the memory cells configured as high-density storage with each cell storing multiple bits of data," *i.e.*, MLC. Dusija, [0109].

Dusija's "first portion" is a "*single level cell (SLC) memory space*": "a memory cell in the first portion is configured to store 1 bit of data as compared to 3 bits of data in the second portion." Dusija, [0109].

d. [1.B] "at least one controller to operate memory elements and associated memory space;"

Dusija discloses this limitation.

Specifically, Dusija discloses controller 102 that is coupled to memory chip 100, which comprises a memory array 200, which in turn comprises the MLC and SLC memory elements (*see* above limitations [1.A.1-2]):



Dusija, Figure 1 (annotated), [0060] (controller "controls" and "manages" the memory operations of memory chip 100, which includes the MLC/SLC memory elements). Dusija also discloses that its controller maintains "block management system" for address mapping. *Id.*, [0117]. Thus, Dusija discloses "*at least one controller* [controller 102] *to operate memory elements* [the MLC and SLC memory elements] *and associated memory space* [the address space for the MLC and SLC memories]." Liu Decl., ¶ 144.

To the extent that Patent Owner argues that the claim term "controller" requires a device external to a flash space, Dusija also discloses this. Dusija, [0062] ("In another embodiment, the controller 102 is implemented within the host."); Liu Decl., ¶ 145.

e. [1.C] "at least one MLC nonvolatile memory element that can be mapped into the MLC memory space;"

Dusija discloses this limitation.

First, Dusija discloses "*at least one MLC nonvolatile memory element*," namely, the MLC memory cells ("memory element[s]") that are in Dusija's "second portion 420" (MLC). *See* limitation [1.A.2].

Second, Dusija discloses that the MLC memory elements "*can be mapped into the MLC memory space*" in two different ways.

First, Dusija's flash memory device uses a "block management system" that "maps and stores [sectors] to the physical structure of the memory array." Dusija, [0060]. Dusija accomplishes this through a "directory . . . embodied in the firmware of the controller" that maps logical addresses to physical addresses. Dusija, [0117]. In other words, Dusija maintains a logical-to-physical address map that maps logical addresses to physical memory elements (MLC or SLC memory elements) in the MLC and SLC physical address spaces (the "section portion 420" and "first portion 410") respectively. For example, Dusija discloses updating the directory to map a logical address to a "second copy" of data written in a "second portion" of memory

(*i.e.*, to MLC memory cells in the MLC memory space) instead of a "first copy" of the data written in a "first portion" of memory (*i.e.*, to SLC memory cells in the SLC memory space). Dusija, [0117]; see also Paley, [0162], Fig. 10B (illustrating such mapping). Accordingly, Dusija discloses that the MLC memory elements "*can be mapped into the MLC memory space*" (Dusija's directory includes an MLC memory space with MLC memory elements that are mapped into this space as part of the logical-to-physical mapping). Liu Decl., ¶¶ 149-51.

Second, Dusija discloses that it configures the memory cells to be either MLC or SLC, which it then maps to the MLC memory space ("second portion 420") and SLC memory space ("first portion 410") respectively. Specifically, Dusija's initial configuration is that the cells are configurable to be either SLC or MLC cells. Dusija, [0059]. It then "[c]onfigur[es] the memory into first [SLC] and second [MLC] portions." *Id.*, [0120], Fig. 14B. Thus, Dusija discloses that the MLC memory elements "*can be mapped into the MLC memory space*" (the memory elements are configured as MLC and SLC and then mapped to the "second portion 420" and "first portion 410" memory spaces, respectively). Liu Decl., ¶¶ 152-54.

f. [1.D] "at least one SLC nonvolatile memory element that can be mapped into the SLC memory space;"

Dusija discloses this limitation.

As discussed above, Dusija discloses "*at least one SLC nonvolatile memory element*," namely, the SLC memory cells ("memory element[s]") that are in Dusija's

"first portion 410" (SLC). *See* limitations [1.A.2], [1.C].

Second, also as explained above (limitation [1.C]), Dusija discloses that the SLC memory elements "*can be mapped into the SLC memory space*" through (1) use of a directory that maps logical addresses to SLC memory elements in the SLC memory space ("first portion 410") (Dusija, [0117], [0129]) and (2) configuring the memory elements to be SLC and assigning them to the SLC space ("first portion 410") (Dusija, [0059], [0109], [0120]); Liu Decl., ¶¶ 156-57.

g. [1.E] "at least one random access volatile memory;"

As discussed above, Dusija discloses or renders obvious "*[a] system comprising . . . at least one random access volatile memory*." *See* limitation [1.A.1] (describing that Dusija would have been understood, or it would have been obvious, for its controller to include RAM and for its cache to be RAM). Liu Decl., ¶ 158.

h. [1.F] "an FTL flash translation layer, wherein the at least one controller, or FTL, or a combination of both maintain an address table in one or more of the memory elements and random access volatile memory"

Dusija discloses, or at least renders obvious, this limitation.[7]

[7] Note that the 300 patent uses "flash translation layer (FTL)" to refer to the well-known concept of translating logical host addresses to physical flash addresses,

First, using a "directory . . . embodied in the firmware of the controller" (Dusija, [0117]), Dusija's flash memory device uses a "block management system" that "maps and stores [sectors] to the physical structure of the memory array" (Dusija, [0060]). In other words, Dusija maintains in its controller a logical-to-physical address map that maps host logical addresses to physical flash memory addresses (the "section portion 420" and "first portion 410"). Sections V.A.5-6; That is, Dusija's controller and directory that map logical to physical addresses is *"an FTL flash translation layer."* Liu Decl., ¶¶ 161-62.

While Dusija does not expressly use the words "FTL," its disclosures would have been understood, and certainly it would have been obvious, to be an FTL. Dusija's directory and controller functionality was well known to operate as a "FTL." Micheloni, 40, 494 ("when the file system sends the command to write or update a certain local page, the FTL actually writes it in a different available physical page and updates a table storing the logical to physical mapping (mapping table)"); Section V.A.6. Moreover, a FTL was conventionally included in flash memory

---

i.e., the functionality that implements the logical-to-physical address map. 300 patent, 2:49-52. That is, the FTL "make[s] linear flash memory [flash addresses] appear to the system like a disk drive [via logical host addresses]." *Id.*, 3:3-6; Liu Decl., ¶ 160.

devices to map host logical addresses to physical flash addresses. Section V.A.6; Liu Decl., ¶¶ 161-62.

Second, Dusija's "directory," *i.e.*, the logical-to-physical address map, is *"an address table."* Dusija, [0117], *see also id.*, [0060]; Liu Decl., ¶ 163.

Third, Dusija maintains its "directory" "in a block management system embodied in the firmware of the controller." Dusija, [0117]. That is, the controller maintains the address table (*"wherein the at least one controller, or FTL, or a combination of both maintain an address table"*). *Id.* Dusija also incorporates by reference Paley (Dusija, [0061]) (providing more details on an exemplary block management system), which teaches that the controller maintains a logical-to-physical memory map to the MLC and SLC memory spaces. Paley, Fig. 10B, [0162], [0171] (pages that includes "SLC" and "MLC"), [0154]-[0155], [0159] (controller maintains map to map host logical addresses to physical addresses); Liu Decl., ¶ 164.

Fourth, Dusija discloses, or at least renders obvious, maintaining its "directory" (*"address map"*) "*in one or more of the memory elements and random access volatile memory*." As Section VII.C.1.b, limitation [1.A.1] illustrates, Dusija would have been understood (or it would have been obvious) to include RAM in its controller. Thus, it would have been understood that Dusija stores the address map in this RAM because, for example, (1) Dusija discloses maintaining its firmware

(software that includes the "directory") in its controller, and (2) Dusija describes its controller managing operations that involve the directory. Dusija, [0059]-[0060], [0117]. Consistent with this, Paley discloses that a logical-to-physical map (*i.e.*, Dusija's directory) "is set up in the controller RAM." Paley, [0021]; Liu Decl. ¶ 165.

To the extent Patent Owner might argue that Dusija's directory must be stored in ***both*** RAM ***and*** non-volatile memory, Dusija discloses or renders obvious that interpretation. As discussed above, Dusija fully incorporates the disclosures of Paley (Dusija, [0061]), which, in turn, incorporates WO 03/027828 to Gorobets et al. (Ex. 1050, "Gorobets WO"), stating that "WO 03/027828 [Gorobets] . . . disclose[s] a memory system . . . [in which] mapping information is buffered in RAM and eventually stored in a sector allocation table in the main memory." Paley, [0023]. In particular, Gorobets PCT discloses storing mapping information in RAM with a "master table" of the mapping information in Flash, which protects against inadvertent power supply loss. Gorobets WO, 26:11-27:21, 2:12-19. Indeed, it was well-known to store the information in RAM (for speed) and Flash (for a back-up). Liu Decl. ¶ 166 (citing Ex. 1033, S. Lee, [0159], [0169], illustrating POSA's background knowledge). Thus, at the very least, a POSA would have been motivated to implement Gorobets' method of storing a mapping information in nonvolatile memory based on Dusija, Paley, and/or the knowledge of a POSA as

routine back-up procedure. Liu Decl. ¶¶ 166-67.

Accordingly, Dusija discloses, or at least renders obvious, that "*wherein the at least one controller, or FTL, or a combination of both* [Dusija's controller implementing the FTL layer] *maintain an address table* [Dusija's directory, *i.e.*, logical-to-physical map] *in one or more of the memory elements* [Dusija' non-volatile memory] *and random access volatile memory* [RAM in Dusija's controller]."

i. [1.G.1] "the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access volatile memory for storage of data therein"

Dusija discloses, or at least renders obvious, this limitation.

First, Dusija discloses that its controller 102 is coupled to memory chip 100, which comprises a memory array 200, which in turn comprises the MLC and SLC elements and its controller manages operations directed to these memories (*"controller controlling access of the MLC and SLC nonvolatile memory elements ... for storage therein"*). Dusija, [0059], [0060], Figure 1; Liu Decl. ¶ 169.

Second, as Section VII.C.1.b, limitation [1.A.1], illustrates, Dusija would have been understood (or it would have been obvious) to include RAM in its controller. It would have been understood, and certainly obvious, that Dusija's controller manages access to this RAM because it is an internal controller component. Moreover, Dusija discloses maintaining its firmware (software that

includes the "directory") in its controller, and Dusija would have been understood to directly access its RAM to execute functions using the directory. Dusija, [0059]-[0060], [0117]. Further, Paley confirms this understanding, as Paley discloses RAM in its controller which the controller accesses. Paley, [0119]. Accordingly, Dusija discloses, or at least renders obvious, *"the controller controlling access of ... the random access volatile memory for storage of data therein."* Liu Decl., ¶ 170.

j. [1.G.2] "the controller, in at least a Write access operation to the MLC nonvolatile memory element, operable to store data in the MLC nonvolatile memory element and retain such stored data in the random access volatile memory;"

Dusija discloses, or at least renders obvious, this limitation. Specifically, Dusija discloses a "post-write read" and "post-write read and adaptive rewrite" that each separately discloses this limitation. Dusija, [0111]-[0116], [0119]-[0124].

First, Dusija discloses "*the controller, in at least a Write access operation to the MLC nonvolatile memory element, operable to store data in the MLC nonvolatile memory element*." In Dusija, the controller writes "incoming data" to the "second portion," *i.e.*, MLC. Dusija, [0111], Section VII.C.1.c, limitation [1.A.2] ("second portion" is MLC); *see also id.*, [0060] (controller manages the memory operations); Liu Decl., ¶ 172.

Second, Dusija discloses "*retain[ing] such stored data in the random access volatile memory*." Dusija explains that once data is written to MLC, the data is read

back and compared to the original copy that had been retained in a "cache." Dusija, [0112]-[0115]. As Section VII.C.1.b, limitation [1.A.1] demonstrates, a POSA would have understood Dusija's teaching of retaining data in a "cache" to disclose retaining the data in a RAM, and thus to "*retain such stored data in the random access volatile memory*." Sections VII.C.1.b and VII.C.1.g demonstrate that at a minimum, it would have been obvious to implement Dusija's "cache" as random access volatile memory. Liu Decl., ¶¶ 173-74.

Dusija discloses this same process as part of a "post-write read and adaptive rewrite" which it depicts in flow-chart form in Figure 15. Dusija, [0119]-[0124].

k. [1.H] "the controller performing a data integrity test on stored data in the MLC nonvolatile memory element after at least a Write access operation performed thereon by comparing the stored data to the retained data in the random access volatile memory;"

Dusija discloses, and at the minimum renders obvious, this limitation.

First, as Section VII.C.1.j, limitation [1.G.2], demonstrates, Dusija discloses a "post-write read" and "post-write read and adaptive rewrite" that writes data to MLC, maintains an original copy in cache, reads back the data from MLC, and compares the data read back to the data in RAM to determine whether the data has a number of bit errors beyond a predetermined amount (*"the controller performing a data integrity test on stored data in the MLC nonvolatile memory element"*). Dusija, [0111]-[0116], [0119]-[0124]. Note that Dusija rewrites the data if it detects

too many errors. *Id.*, [0116], [0124]; Liu Decl., ¶ 176.

Second, in Dusija, the "post-write read" and "post-write read and adaptive rewrite" occur after the write operation ("*after at least a Write access operation performed thereon*"). Dusija, [0112] ("Later, the first copy of the data page is read back"), [0122] (reading back "after a predefined time"). Indeed, Dusija describes the test as involving a "***post-write***" operation. Liu Decl., ¶ 177.

Third, Dusija would have been understood to disclose that the copy of data read back from MLC is stored with the original copy in volatile memory (or at a minimum it would have been obvious). Recall that data is retained in Dusija's "cache," which would have been understood, and at least obvious, to be the controller's RAM. *See* Section VII.C.1.b. As Paley, Micheloni, and Harari illustrate, a controller typically has a RAM as a cache and to support other processing. *Id.* Indeed, it was well known for a controller to use such RAM for such operations, i.e., Dusija's data integrity test. *Id.*; Section V.A.4, VII.C.1.b, VII.C.1.g; Liu Decl. ¶ 178.

Dusija's data integrity test may be performed after each write. Dusija, [0203]; Liu Decl. ¶ 179.

Thus, Dusija discloses performing the data integrity test "*by comparing the stored data to the retained data in the random access volatile memory*." *See* Dusija, [0112] ("post write read" and "comparison with the original copy which may be

cached"), [0122]-[0023]; Liu Decl. ¶ 180. And the controller, using its RAM, would be understood to perform this operation. *See* Section VII.C.1.i (describing controller's RAM and control thereof); Dusija, Fig. 1, [0062]; Liu Decl. ¶ 181.

l. [1.I] "wherein the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and"

Dusija discloses this limitation.

First, Dusija discloses a "directory" that is a logical-to-physical address table and that is adaptable to cover the specific "first portion" and "section portion" system configuration, *i.e.*, it maps the logical addresses to the configured MLC and SLC system physical addresses ("*wherein the address table maps logical and physical addresses adaptable to the system*"). *See* Section VII.C.1.e, h, limitations [1.C, F]; Liu Decl., ¶ 183.

Second, regarding "*maximiz[ing] lifetime*," the 300 patent describes a data integrity test in which if a write failure to MLC occurs, the data is rewritten to more robust SLC cells to "boost lifetime (endurance)." 300 patent, 5:14-20; *see also id.*, 1:45-52 (increasing lifetime by using MLC and SLC); Liu Decl., ¶ 184.

Just like the 300 patent, as Section VII.C.1.m, limitation [1.J] (*infra*), demonstrates, Dusija discloses remapping data from MLC to SLC memory if a write to MLC fails ("*the mapping is performed as necessitated by the system to maximize*

lifetime"). MLC is less robust (more error prone) than SLC, and thus if writes fail to MLC, the data can be written to more robust SLC. Dusija, Abstract, [0109], [0117]. SLC provides a portion of memory that will last longer than MLC and thus writing to SLC when MLC fails, as Dusija discloses, will maximize lifetime. *Id.*; *see also id.*, [0110] (recognizing that SLC cells provide for "longer operational life"); Liu Decl. ¶ 185.

Note that Dusija discloses other mappings techniques that increase memory lifetime (*"the mapping is performed as necessitated by the system to maximize lifetime"*). For example, Dusija discloses remapping a logical address among different MLC blocks as MLC blocks fail, and performing such remapping until all MLC blocks fail. Dusija, [0162]. By way of another example, Dusija discloses attempting first writing data from the host to SLC (the more robust memory) and then attempting to write it to MLC (less robust), but if the MLC blocks fail, maintaining the logical to physical mapping to SLC. Dusija, [0166]-[0174]. These techniques maximize lifetime by using alternative memory locations as memory elements fail. Liu Decl. ¶ 186.

Third, Dusija's "directory" is a logical-to-physical address table. *See* Section VII.C.1.e, h, limitations [1.C, F]. Dusija discloses that the mapping may include "virtual [] blocks" (*i.e.*, *"blocks"*), Dusija, [0167], and "page[s]" in memory (*i.e.*, *"pages"*), [0012] ("multi-sector page"), [0111], [0117] (incoming logical pages

written to physical MLC/SLC pages). Paley also discloses block-level addressing. Paley, [0016] ("logical blocks"), [0163] ("logical address LBA") (*i.e.*, *"blocks"*). Thus, Dusija discloses *"wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories."* Liu Decl., ¶ 187.

m. [1.J] "wherein a failure of the data integrity test performed by the controller results in a remapping of the address space to a different physical range of addresses and transfer of data corresponding to the stored data to those remapped physical addresses from those determined to have failed the data integrity test to achieve enhanced endurance."

Dusija discloses this limitation.

First, Dusija discloses that data is first written to the "second portion" (*i.e.*, MLC, Dusija, [0109]). Dusija, [0111]. Then, the data is read back from MLC for a comparison with an original cache copy to check for errors. *Id.*, [0112]. Upon a failure of the comparison, *i.e.*, when "the number of error bits in the data page has exceeded the predetermined amount," "the data page is rewritten to the first portion" (*i.e.*, SLC, *id.*, [0109]). *Id.*, [0116]. The controller then remaps the logical addresses to the new SLC physical addresses (to which the data was "rewritten"), which Dusija refers to as "updat[ing]" the "directory in a block management system." *Id.*, [0117]. Thus, Dusija discloses *"wherein a failure of the data integrity test performed by the controller* [the read and comparison] *results in a remapping of the address space to*

a different physical range of addresses [remapping the logical address from MLC to SLC physical address ranges]*."*[8] Liu Decl., ¶ 189.

Second, the above process also results in the *"transfer of data corresponding to the stored data to those remapped physical addresses from those determined to have failed the data integrity test to achieve enhanced endurance."* Specifically, Dusija discloses writing the cache copy (*"data corresponding to the stored data"*) to the SLC physical address ("*to those remapped physical addresses"*). Dusija, [0116] (rewrite), [0117] (remapping). Thus, as part of this rewrite, the data is transferred to *"those remapped physical addresses"* (the new SLC addresses) "*from those determined to have failed the data integrity test*" (from the MLC addresses that failed the test). Finally, as Section VII.C.1.l, limitation [1.I] demonstrates, this process is "*to achieve enhanced endurance."* In particular, because MLC cells have less endurance (and fail faster) than SLC cells, rewriting the data from MLC to SLC will achieve greater endurance. *See* Section VII.C.1.l; Liu Decl., ¶¶ 190-91.

Dusija discloses this same process as part of a "post-write read and adaptive rewrite." Dusija, [0119]-[0124].

---

[8] The 300 patent uses "address ranges" to refer to basic flash addressing, *e.g.*, a block, page, or sector address. 300 patent, Figures 2a-b, 5:54-58 (explaining minimum quanta of address, *e.g.*, a block or page), 3:7-11 (sector); Liu Decl., fn. 8.

2. Claim 2: "The system of claim 1, wherein the FTL flash translation layer is a software module, or a firmware module containing software updates."

Dusija discloses, or at least renders obvious, this limitation by the disclosure of its "directory," which is a flash translation layer. *See* Section VII.C.1.h, limitation [1.F]. Further, Dusija discloses that its "directory" is "embodied in the firmware of the controller." *E.g.*, Dusija, [0117]. Firmware, by definition, is software (thus the firmware is a software module). Ex. 1054, Microsoft Computer Dictionary, 215. Paley also discloses that its "memory manager . . . is a software component." Paley, [0153]. Thus, Dusija's "directory" (*"FTL flash translation layer"*) would be understood to be a *"software module."* Liu Decl., ¶ 192.

3. Claim 3: "The system of claim 1, wherein at least one of the random access volatile memory or the MLC and SLC nonvolatile memory elements are embedded in the at least one controller."

Dusija discloses, or at least renders obvious, this limitation.[9]

As Section VII.C.1.b, limitation [1.A.1], demonstrates, Dusija discloses or renders obvious "*random access volatile memory*" being "*embedded*" in the

[9] The 300 patent does not describe what it means for memory to be "embedded" in the controller other than mentioning a controller containing memory. 300 patent, 6:15-32; Liu Decl., ¶ 194.

"*controller.*" As that section describes, Dusija discloses controller 102, which includes firmware 60 that manages memory operations. Dusija, Fig. 1, [0062]. A POSA would have understood, and at least found it obvious, that Dusija's controller includes RAM to temporarily store the firmware during execution and temporarily store data used during memory operations. And Dusija fully incorporates the disclosures of Paley, which discloses a controller for a flash memory device that includes RAM. Paley, Fig. 1, [0119]. Embedding RAM into a controller was conventional. *Id.*; Liu Decl., ¶ 195.

4. Claim 4: "The system of claim 1, wherein the MLC and SLC nonvolatile memory elements comprise flash memory."

Dusija discloses this limitation, namely, that the MLC and SLC non-volatile memory elements are flash memory. Dusija, [0018] ("flash memory"), [0059], [0068], [0109] (MLC and SLC); Liu Decl., ¶ 196.

5. Claim 5: "The system of claim 1, wherein the random access volatile memory is dynamic random access memory."

Dusija discloses, or at least renders obvious, this limitation.

As Section VII.C.1.b, limitation [1.A.1], demonstrates, Dusija discloses or render obvious "*random access volatile memory*" for use as Dusija's cache. As Section V.A.4 demonstrates, the two conventional choices for such caches were DRAM and SRAM. As such, a POSA would have understood Dusija's disclosure of a "cache" to include a DRAM cache as an option. Indeed, during the prosecution

of the 300 patent, the Examiner repeatedly took official notice that SRAM and DRAM are "volatile memory types" that are often used as "cache memory." Ex. 1008, 48, 78. The Applicant did not contest this use of official notice. At a minimum, it would have been obvious for Dusija to use DRAM for its cache given that DRAM was conventionally used as a cache (*see* Section V.A.4), DRAM was known to have advantages as a cache over other memories (including memory density and cost), and using DRAM as a cache was highly predictable. Liu Decl., ¶ 198.

6. Claim 6: "The system of claim 1, wherein the random access volatile memory is static random access memory."

Dusija discloses, or at least renders obvious, this limitation.

As Section VII.C.1.b, limitation [1.A.1], demonstrates, Dusija discloses or renders obvious "*random access volatile memory*" for use as Dusija's cache. As Section V.A.4 demonstrates, the two conventional choices for such caches were DRAM and SRAM. As such, a POSA would have understood Dusija's disclosure of a "cache" to include a SRAM cache as an option. Dusija also incorporates by reference Paley, which discloses using SRAM as a volatile memory. Paley, [0423]-[0425], [0502]-[0503]; Liu Decl., ¶ 200.

Moreover, the Examiner repeatedly took official notice that SRAM and DRAM are "volatile memory types" that are often used as "cache memory." Ex. 1008, 48, 78. At a minimum, it would have been obvious for Dusija to use

SRAM for its cache given that SRAM was conventionally used as a cache (*see* Section V.A.4), SRAM was known to have advantages as a cache over other memories (including high speed operation), and using SRAM as a cache was highly predictable. Liu Decl., ¶ 201.

7. Claim 7: "The system of claim 1, wherein the controller, upon detection of a failure of the data integrity test, remaps the data to the SLC nonvolatile memory element."

Dusija discloses this limitation.

As Section VII.C.1.m, limitation [1.J], demonstrates, Dusija discloses a data integrity test "*wherein the controller, upon detection of a failure of the data integrity test, remaps the data to the SLC nonvolatile memory element*." In particular, when a data integrity test fails, Dusija discloses rewriting the data to the "first portion" (which is SLC, Dusija, [0109]) and updating the address map in the controller ("updat[ing]" the "directory"). Dusija, [0116]-[0117]; *see also id.*, [0124]-[0125]. A POSA would understand that Dusija's controller performs these operations. *Id.*, [0060] (the controller "controls and manages higher level memory operations."). Liu Decl., ¶ 203.

8. Claim 8: "The system of claim 7, wherein the SLC memory element has a higher endurance than the MLC memory element."

Dusija discloses this limitation.

Dusija discloses that memory cells operating as SLC have a "longer operational life," *i.e.*, *"higher endurance,"* than MLC memory cells. Dusija, [0110]; *see also* 300 patent, 3:41-45 (SLC is known to have higher endurance); Liu Decl., ¶ 205.

9. Claim 9: "The system of claim 1, wherein the MLC is a multilevel cell, wherein the multilevel cell stores at least 2 bits per cell."

Dusija discloses this limitation, namely, "[t]he second portion 420 [MLC] … with each cell storing multiple bits of data." Dusija, [0109]. Dusija's MLC is multilevel cell memory. *Id*., [0059]; Liu Decl., ¶ 206.

10. Claim 11: "The system of claim 1 wherein the MLC allows a single cell to store multiple bits."

Dusija discloses this limitation, namely, "[t]he second portion 420 [MLC] … with each cell storing multiple bits of data." Dusija, [0109]. Dusija's MLC is multilevel cell memory. *Id*., [0059]; Liu Decl., ¶ 207.

11. Claim 12:

a. [12.PRE] "A system for storing data comprising:"

To the extent the preamble is limiting, the preamble is identical to, and satisfied for the same reasons, as [1.PRE]. *See* Section VII.C.1.a.

b. [12.A] "memory space containing volatile memory space and nonvolatile memory space, wherein the nonvolatile memory space includes both multilevel cell (MLC) space and single level cell (SLC) space;"

This limitation is satisfied for the same reasons as, and is effectively identical to, limitations [1.A.1-2] (this limitation does not include "memory" after "(MLC)" and "(SLC)"). *See* Sections VII.C.1.b-c.

c. [12.B] "at least one controller to operate memory elements and associated memory space, and to maintain an address table in one or more of the memory elements;"

This limitation is satisfied for the same reasons as, and is identical to, limitations [1.B] and [1.F] (a portion of 1.F). *See* Sections VII.C.1.d, h.

d. [12.C] "at least one MLC nonvolatile memory element that can be mapped into the nonvolatile memory space;"

e. [12.D] "at least one SLC nonvolatile memory element that can be mapped into the nonvolatile memory space;"

These limitations are satisfied for the same reasons as, and are nearly identical to, limitations [1.C-D]. *See* Sections VII.C.1.e-f. These limitations require mapping "into the nonvolatile memory space." Limitations 1.C-D requires mapping into "MLC" and "SLC memory space," which is "nonvolatile memory space." Liu Decl., ¶ 211.

f. [12.E] "at least one random access volatile memory;"

This limitation is satisfied for the same reasons as, and is identical to, limitation [1.E]. *See* Section VII.C.1.g.

g. [12.F] "the controller controlling access of the MLC and SLC nonvolatile memory elements and the random access volatile memory for storage of data therein, the controller, in at least a Write access operation to the MLC nonvolatile memory element, operable to store data in the MLC nonvolatile memory element and retain such stored data in the random access volatile memory;"

This limitation is satisfied for the same reasons as, and is identical to, limitations [1.G.1-2]. *See* Sections VII.C.1.i-j.

h. [12.G] "the controller performing a data integrity test on stored data in the MLC nonvolatile memory element after at least a Write access operation performed thereon by comparing the stored data to the retained data in the random access volatile memory;"

This limitation is satisfied for the same reasons as, and is identical to, limitation [1.H]. *See* Section VII.C.1.k.

i. [12.H] "wherein the address table maps logical and physical addresses adaptable to the system, wherein the mapping is performed as necessitated by the system to maximize lifetime, and wherein the mapping maps blocks, pages, or bytes of data in either volatile or nonvolatile, or both, memories; and"

This limitation is satisfied for the same reasons as, and is identical to, limitation [1.I]. *See* Section VII.C.1.l.

j. [12.I] "wherein a failure of the data integrity test performed by the controller results in a remapping of the address space to a different physical range of addresses and transfer of data corresponding to the stored data to those remapped physical addresses from those determined to have failed the data integrity test to achieve enhanced endurance."

This limitation is satisfied for the same reasons as, and is identical to, limitation [1.J]. *See* Section VII.C.1.m.

D. Ground 2: Dusija In View Of Sutardja And The Knowledge Of A POSA Renders Obvious Claim 10

Claim 10 depends on claim 1. As demonstrated above, claim 1 is obvious in view of Dusija. Although Claim 10 is not rendered obvious by Dusija alone, it is obvious in view of Dusija in combination with Sutardja.

1. Claim 10: "The system of claim 1, wherein the contents of frequently accessed portions of the memory space are periodically moved from the MLC space to the SLC space."

For the Dusija and Sutardja disclosures of claim 1, see Section V.II.C.

Dusija in view of Sutardja renders this limitation obvious.

Specifically, Sutardja discloses a "data shift analysis." Sutardja, [0148]-[0149]. When the time to perform data shift analysis has arrived, Sutardja discloses determining if "a number of write operations to a first block of the first NVS memory [*i.e.*, MLC] during a predetermined time is greater than or equal to a predetermined threshold" (*"contents of frequently accessed portions of the memory space"*). *Id.*, [0149], Figure 7C, [0106], claim 37 (first memory is MLC; second memory is SLC).

If greater than the threshold, the "control maps the logical addresses that correspond to the first block to a second block of the second NVS memory [*i.e.*, SLC memory module] in step 522" (*"moved from the MLC space to the SLC space"*). *Id.* Note that by Sutardja disclosing remapping the logical address to SLC, it would have been understood that it must move the data to SLC as well, which Sutardja describes in paragraph 167 ("swapping data"). *Id.*; Liu Decl., ¶ 220.

Note that Sutardja's "data shift" involves moving the contents of MLC blocks to SLC blocks without the data changing, *i.e.*, not in connection to with a host write operation. Sutardja, [0148]-[0149]. In other words, Sutardja discloses moving the data from MLC memory modules to SLC memory modules. *Id.*, [0148]-[0149]; *see also id.*, [0167] (describing "swapping data"); Liu Decl., ¶ 221.

In addition, and unlike the "data shift," Sutardja discloses a second way to use the frequency count to move data from MLC memory to SLC memory. Specifically, upon receiving a host write to an MLC block, if the frequency count to that logical address is high, it may direct the write to a physical SLC block, thereby transferring the data from MLC memory modules to SLC memory modules. Sutardja, [0146]-[0147]; Liu Decl., ¶ 222.

As discussed above, Sutardja discloses that its system "determines whether [the] time to perform data shift analysis has arrived in step 514." Sutardja, [0148].

In other words, the move occurs periodically, *e.g.*, when the device is not otherwise busy. *Id.*; Liu Decl., ¶ 223.

Similarly, Sutardja discloses that "[a]t various times, ***such as periodically***, the wear leveling module may analyze the wear levels of the blocks, ***and*** remap relatively frequently rewritten logical addresses to blocks with low wear levels." Sutardja, [0167]. The data analysis is a wear-leveling technique. *Id.*, [1048]. Thus, Sutardja expressly discloses performing its content transferring techniques *"periodically."* *Id.* A POSA would have understood, and at least found obvious, that Sutardja is teaching that its content moving techniques are applicable periodically. Liu Decl., ¶ 223. Indeed, as Dr. Liu explains, it was typical to perform such transfers according to an overall access count (consistent with whether time has arrived). Liu Decl., ¶ 224 (citing Ex. 1030, "Chin," [0018], [0074] and citing Ex. 1031, "Radke," [0022], [0026] as evidence of knowledge of a POSA).

2. Motivation to Combine

A POSA would have considered it obvious and would have been motivated to combine the teachings of Dusija and Sutardja to arrive, with a reasonable expectation of success, at the subject matter of claim 10. Liu Decl., ¶ 226.

First, both Dusija and Sutardja describe techniques for improving the performance of flash memory systems that include both MLC and SLC memory. Dusija, [0018]; Sutardja, Abstract, [0012]. Thus, a POSA with knowledge of

Dusija's system would have looked to Sutardja for other techniques that take advantage of the differences between MLC and SLC. Liu Decl., ¶ 227.

Second, Sutardja's rationale for periodically remapping hot blocks (frequently written MLC blocks) to SLC—to optimize cost and memory lifetime (Sutardja, [0102])—applies equally to Dusija's system that also seeks to maximize memory lifetime. That is, by allocating frequently written blocks to SLC in Dusija's system, a POSA would have understood that this would optimize cost and memory lifetime by allocating frequently written blocks to the type of cells that can handle more wear (and better handle errors). Dusija, [0018]; Section V.A.2, 6; Liu Decl., ¶ 228.

Third, a POSA would have understood that mapping higher-usage logical addresses to SLC would improve memory access time, because access time to SLC cells was faster than MLC cells. Micheloni, 511. Thus, a POSA would have recognized that integrating Sutardja's teachings into Dusija would further improve the speed of Dusija's system. Liu Decl., ¶ 229.

Fourth, a POSA would have been motivated to allocate the frequently written data to an SLC memory module, which has a faster access time and can handle more wear, and the more static data to an MLC memory module that is more dense and less expensive. Ex. 1019, Micheloni at 511; Liu Decl., ¶ 230.

Fifth, a POSA would have had a high expectation of success in adapting Dusija to include Sutardja's hot blocks teaching. To start, Dusija already discloses

maintaining an erase count for blocks ("hot count"), illustrating that such counting was routine and applicable to Dusija, and discloses. Dusija, [0101], [0154]. And Dusija states that other features and modifications are implementable within its system. *Id.*, [0151]; Liu Decl., ¶ 231. Further, Dusija already discloses memory management features that occur in the background, *e.g.*, counting erase cycles and enabling error management. *Id.*, [0155]-[0160]. Thus, Dusija's system is compatible with implementing Sutardja's hot block allocating feature in its background processing. Finally, as Sections VII.C.1.k-m explain, Dusija discloses rewriting data and remapping the logical address to a new physical address as part of a data integrity test. Sutardja similarly discloses rewriting data and remapping addresses, but in the context of allocating frequently written blocks. Thus, Dusija already includes the building blocks for rewriting and remapping data to carry out Sutardja's allocation of frequently written blocks. Liu Decl., ¶ 232.

VIII. CONCLUSION

For the foregoing reasons, claims 1-12 are unpatentable. Accordingly, Petitioner respectfully requests institution of IPR and cancellation of these claims.

Dated: September 21, 2021 ORRICK, HERRINGTON & SUTCLIFFE LLP



By: _____

Jeremy Jason Lang
Lead Counsel for Petitioner
Reg. No. 73,064
1000 Marsh Road
Menlo Park, CA 94025-1015
T: (650) 614-7400
F: (650) 614-7401
Email: PTABDocketJJL2@orrick.com

Christopher Childers
Reg. No. 75,237
Columbia Center
1152 15th Street, N.W.
Washington, D.C. 20005
T: (202) 339 8400
F: (202) 339 8500
Email: PTABDocketC4C8@orrick.com

Parth Sagdeo
Reg. No. 71,275
222 Berkeley Street, Suite 2000
Boston, MA 02116
T: (617) 880 1800
F: (617) 880 1801
Email: PTABDocketP2S7@orrick.com

Jared Bobrow
Pro Hac Vice to be submitted
1000 Marsh Road
Menlo Park, CA 94025-1015
T: (650) 614-7400
F: (650) 614-7401
Email: PTABDocketJ3B3@orrick.com
Attorneys for Petitioner Micron Technology, Inc.

CERTIFICATION PURSUANT TO 37 C.F.R § 42.11

Pursuant to 37 C.F.R. § 42.11, the undersigned hereby certifies that this Petition and accompanying evidence including Exhibits 1001-1054 are not being presented for an improper purpose and that all legal contentions, allegations, and denials are warranted and have evidentiary support.

Dated: September 21, 2021 ORRICK, HERRINGTON & SUTCLIFFE LLP

By: 

Jeremy Jason Lang
Lead Counsel for Petitioner
Reg. No. 73,064
1000 Marsh Road
Menlo Park, CA 94025-1015
T: (650) 614-7400
F: (650) 614-7401
Email: PTABDocketJJL2@orrick.com

CERTIFICATE OF COMPLIANCE

The undersigned certifies that the foregoing PETITION FOR *INTER PARTES* REVIEW complies with the type volume limitation in 37 C.F.R. § 42.24(a)(1)(i). According to the utilized word-processing system's word count, the petition—excluding the caption, table of contents, table of exhibits, mandatory notices, certificate of word count, and certificate of service—contains 13,516 words.

Dated: September 21, 2021 ORRICK, HERRINGTON & SUTCLIFFE LLP

By: 

Jeremy Jason Lang
Lead Counsel for Petitioner
Reg. No. 73,064
1000 Marsh Road
Menlo Park, CA 94025-1015
T: (650) 614-7400
F: (650) 614-7401
Email: PTABDocketJJL2@orrick.com

CERTIFICATE OF SERVICE

The undersigned hereby confirms that the foregoing Petition for *Inter Partes* Review and associated exhibits were caused to be served on September 22, 2021 via overnight mail upon the following counsel of record for Patent Owner:

MUNCK WILSON MANDALA L.L.P
P.O. DRAWER 800889
DALLAS TX 75380

MUNCK WILSON MANDALA L.L.P
600 BANNER PLACE TOWER
12770 COIT RD
DALLAS, TX 75251
P: 972.628.3600

Copies of this Petition and accompanying exhibits were also served on September 21, 2021, via electronic mail on Patent Owner's counsel of record in a related district court litigation:

Alan L. Whitehurst (awhitehurst@McKoolSmith.com)
Christopher P. McNett (cmcnett@McKoolSmith.com)
Kathy H. Li (kli@McKoolSmith.com)
James E. Quigley (jquigley@McKoolSmith.com)
MCKOOL SMITH, P.C.

/*Karen Johnson*/
Karen Johnson