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

DECLARATION OF DR. DAVID LIU

TABLE OF CONTENTS **Page**

I. INTRODUCTION 8
II. EDUCATION BACKGROUND, PROFESSIONAL EXPERIENCE, AND OTHER QUALIFICATIONS 8
III. ASSIGNMENT AND MATERIALS CONSIDERED 11
IV. UNDERSTANDING OF THE LAW 14
V. LEVEL OF SKILL IN THE ART 18
VI. THE 300 PATENT'S EFFECTIVE FILING DATE 19
VII. THE 300 PATENT 19
A. Technological Background 19
1. Volatile, Non-volatile, and Flash Memory 19
2. Programming Flash, and SLC and MLC Flash Memory Cells 21
3. Flash Architecture 23
4. Caching 26
5. Logical Addresses, Physical Addresses, Bad Block Replacement, and Wear Leveling 27
6. Flash Translation Layer ("FTL") 36
7. Speed and Durability Considerations for MLC and SLC Cells 40
8. Data Integrity Tests 41
B. Summary of the 300 Patent's Disclosure 41
C. The 300 Patent's Prosecution History 44
VIII. CLAIM CONSTRUCTION 48
A. "data integrity test" (claims 1 and 12) 49
B. "comparing the stored data to the retained data in the random access volatile memory" (claims 1 and 12) 50
C. "periodically" 53
D. Other Terms 54

Vervain Ex. 2011 p. 2 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 2
Micron v. Vervain
IPR2021-01550

IX. HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE 54
A. Prior Art Overview 54
1. Dusija 54
2. Sutardja 57
B. Ground 1: Dusija In View Of The Knowledge Of A POSA Renders Obvious Claims 1-9 And 11-12 59
1. Claim 1 59
a. [1.PRE] "A system for storing data comprising:" 59
b. [1.A.1] "memory space containing volatile memory space and nonvolatile memory space;" 60
c. [1.A.2] "wherein the nonvolatile memory space includes both multilevel cell (MLC) memory space and single level cell (SLC) memory space;" 66
d. [1.B] "at least one controller to operate memory elements and associated memory space;" 68
e. [1.C] "at least one MLC nonvolatile memory element that can be mapped into the MLC memory space;" 70
f. [1.D] "at least one SLC nonvolatile memory element that can be mapped into the SLC memory space;" 74
g. [1.E] "at least one random access volatile memory;" 75
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" 76
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" 80

Vervain Ex. 2011 p. 3 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 3
Micron v. Vervain
IPR2021-01550

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;".....82

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;" 83

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 volaile or nonvolatile, or both, memories; and".....85

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 addreses from those determined to have failed the data integrity test to achieve enhanced endurance.".....88

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.".....91

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." 91

4. Claim 4: "The system of claim 1, wherein the MLC and SLC nonvolatile memory elements comprise flash memory.".....92

Vervain Ex. 2011 p. 4 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 4
Micron v. Vervain
IPR2021-01550

5. Claim 5: "The system of claim 1, wherein the random access volatile memory is dynamic random access memory."....92

6. Claim 6: "The system of claim 1, wherein the random access volatile memory is static random access memory."....93

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."....94

8. Claim 8: "The system of claim 7, wherein the SLC memory element has a higher endurance than the MLC memory element."....95

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."....95

10. Claim 11: "The system of claim 1 wherein the MLC allows a single cell to store multiple bits."....95

11. Claim 12:....96

    a. [12.PRE] "A system for storing data comprising:"....96

    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;"....96

    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;"....96

    d. [12.C] "at least one MLC nonvolatile memory element that can be mapped into the nonvolatile memory space;"....97

    e. [12.D] "at least one SLC nonvolatile memory element that can be mapped into the nonvolatile memory space;"....97

Vervain Ex. 2011 p. 5 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 5
Micron v. Vervain
IPR2021-01550

f. [12.E] "at least one random access volatile memory;" 97

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;".....97

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;" 98

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".....98

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.".....98

C. Ground 2: Dusija In View Of Sutardja And The Knowledge Of A POSA Renders Obvious Claim 10 98

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." 99

2. Motivation to Combine 103

Vervain Ex. 2011 p. 6 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 6
Micron v. Vervain
IPR2021-01550

X. OBJECTIVE INDICIA OF NON-OBVIOUSNESS.....106
XI. DECLARATION.....107

Vervain Ex. 2011 p. 7 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 7
Micron v. Vervain
IPR2021-01550

I, Dr. David Liu, hereby declare as follows:

I. INTRODUCTION

1. I have been retained by Micron Technology, Inc. ("Micron") as an independent expert consultant in this proceeding before the United States Patent and Trademark Office ("PTO"). I am not an employee of Micron or any affiliate or subsidiary of Micron.

2. I have been asked to consider whether certain references teach or suggest the features recited in certain claims of U.S. Patent No. 10,950,300, which I refer to herein as the 300 Patent, and whether certain claims of the 300 Patent are unpatentable as obvious.

3. My opinions and the bases for my opinions are set forth below.

4. I am being compensated at $550 per hour for my work, plus reimbursement for any reasonable expenses. My compensation is based solely on the amount of time that I devote to activity related to this case and is in no way contingent on the nature of my findings, the presentation of my findings in testimony, or the outcome of this or any other proceeding. I have no other financial interest in this proceeding.

II. EDUCATION BACKGROUND, PROFESSIONAL EXPERIENCE, AND OTHER QUALIFICATIONS

5. My curriculum vitae ("CV") is attached hereto as Attachment A and provides an accurate identification of my background and experience. Among

Vervain Ex. 2011 p. 8 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 8
Micron v. Vervain
IPR2021-01550

other things, my CV includes a list of my issued patents and publications for at least the last 30 years. My curriculum vitae also includes information about my expert witness experience, with case numbers where available.

6. I received a Bachelor of Science in Electrical Engineering from the University of California, Berkeley, in 1983. I received a Master of Science and Ph.D. degrees in Electrical Engineering from Stanford University in 1985 and 1989, respectively.

7. From 1989 to 1992, I was a member of technical staff at Texas Instruments, Inc. At Texas Instruments, my job responsibilities included process integration, device modeling, high-voltage CMOS process integration, and investigating novel source-side injection mechanisms for Flash EPROM channel hot-electron programming.

8. From 1992 to 1995, I was a member of the technical staff at Advanced Micro Devices, Inc. (AMD) where I was a key contributor in optimizing Flash cell and periphery devices in AMD's CMOS-based 0.5um and 0.35um Flash EPROM technology. My job responsibilities at AMD also included process integration, device modeling, and development of triple-well process technology for x-decoder transistors and high voltage transistors for negative gate erase operation. While at AMD, I was awarded a Spotlight Award for developing a method of manufacturing a self-aligned source (SAS) etch for a NOR flash memory.

Vervain Ex. 2011 p. 9 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 9
Micron v. Vervain
IPR2021-01550

9. I spent the next five years of my career in managerial and director roles at several California-based semiconductor companies. I was responsible for increasing yield and for leading teams of engineers working to develop next-generation memory devices.

10. In 2000, I co-founded Progressant Technologies in Fremont, California. Progressant Technologies developed IP for negative differential resistance transistor technology and was eventually acquired by Synopsys, Inc.

11. From 2000 to 2004, I was a Senior Manager at Xilinx, Incorporated, where I was responsible for developing nonvolatile memory process technology for flash and CPLD product applications, as well as advanced CMOS process technology (specifically 75nm CMOS technology node, a half node version between 90nm and 65nm).

12. From 2004 to 2007, I was a Senior Scientist at Maxim Integrated Products where I was responsible for developing Embedded Non-volatile Memory process technology for Power Management product applications.

13. Since 2007, I have served as a technical consultant where I have provided expert advice regarding Flash memory technology, CMOS process technology, and semiconductor device physics.

14. I am not now and have never been an employee of the Petitioner.

Vervain Ex. 2011 p. 10 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 10
Micron v. Vervain
IPR2021-01550

III. ASSIGNMENT AND MATERIALS CONSIDERED

15. I have been asked to provide analysis and explain the subject matter of the 300 Patent, including the state of the art when the 300 Patent application was filed. I have also been asked to consider, analyze, and explain certain prior art to the 300 Patent including how that art relates to the challenged claims of the 300 Patent and to provide my opinions regarding whether that art invalidates the claimed subject matter.

16. The opinions expressed in this declaration are not exhaustive of my opinions regarding the unpatentability of the claims of the 300 Patent. Therefore, the fact that I do not address a particular point should not be understood to indicate an agreement on my part that any claim complies with the requirements of any applicable patent or other rule.

17. I reserve the right to amend and supplement this declaration in light of additional evidence, arguments, or testimony presented during this IPR or related proceedings on the 300 Patent.

18. In forming the opinions set forth in this declaration, I have considered and relied upon my education, knowledge of the relevant field, knowledge of scientific and engineering principles, and my experience. I have also reviewed and considered the 300 Patent (Exhibit 1007), its prosecution history (Exhibit 1008), and the following additional materials:

Vervain Ex. 2011 p. 11 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 11
Micron v. Vervain
IPR2021-01550

Exhibit	Description
1010	U.S. Patent Application Publication No. 2011/0099460 ("Dusija")
1011	U.S. Patent Application Publication No. 2008/0140918 ("Sutardja")
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")
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

Vervain Ex. 2011 p. 12 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 12
Micron v. Vervain
IPR2021-01550

Exhibit	Description
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")
1033	U.S. Patent Application Publication No. 2006/0053246 ("S. Lee")
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")
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 of address space
1052	U.S. Patent Application Publication No. 2009/0300269 ("Radke Appl.")
1053	U.S. Patent No. 8,250,333 ("Gorobets II")

Vervain Ex. 2011 p. 13 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 13
Micron v. Vervain
IPR2021-01550

Exhibit	Description
1054	Microsoft Computer Dictionary, Fifth Edition, 2002, definition of firmware

IV. UNDERSTANDING OF THE LAW

19. I am not an attorney, but I have been instructed in and applied the law as described in this section.

20. I understand that the first step in comparing an asserted claim to the prior art is for the claim to be properly construed. I address how a person of ordinary skill in the art would have understood the claims of the alleged invention in Section IX below.

21. I have been further instructed and understand that a patent claim is unpatentable and invalid as obvious if the subject matter of the claim as a whole would have been obvious to a person of ordinary skill in the art of the claimed subject matter as of the time of the invention at issue. I understand that when assessing the obviousness of claimed subject matter, the following factors are evaluated: (1) the scope and content of the prior art; (2) the difference or differences between each claim of the patent and the prior art; and (3) the level of ordinary skill in the art at the time the patent was filed.

22. I understand that claimed subject matter may be obvious in view of more than one item of prior art. I understand, however, that it is not enough to show simply that all the limitations of the claimed subject matter are spread

Vervain Ex. 2011 p. 14 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 14
Micron v. Vervain
IPR2021-01550

throughout the prior art. Instead, for claimed subject matter to be obvious over multiple references, there must be some reason or motivation for one of ordinary skill in the art to combine the prior art references to arrive at the claimed subject matter.

23. I have been informed that, in seeking to determine whether an invention that is a combination of known elements would have been obvious to a person of ordinary skill in the art at the time of the invention, one must consider the references in their entirety to ascertain whether the disclosures in those references render the combination obvious to such a person.

24. I have been informed and understand that, while not required, the prior art references themselves may provide a teaching, suggestion, motivation, or reason to combine, but other times the motivation linking two or more prior art references is common sense to a person of ordinary skill in the art at the time of the invention.

25. I understand that a particular combination may be proven obvious merely by showing that it was obvious to try the combination. I have been informed that, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

Vervain Ex. 2011 p. 15 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 15
Micron v. Vervain
IPR2021-01550

26. I further understand that an obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references also may be supplied by the direction of the marketplace. For example, when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp because the result is likely the product not of innovation but of ordinary skill and common sense.

27. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. Thus, where all of the elements of a claim are used in substantially the same manner, in devices in the same field of endeavor, the claim is likely obvious.

28. Additionally, I understand that a patent is likely to be invalid for obviousness if a person of ordinary skill can implement a predictable variation or if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. Therefore, when a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one.

Vervain Ex. 2011 p. 16 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 16
Micron v. Vervain
IPR2021-01550

29. I further understand that combining embodiments related to each other in a single prior art reference would not ordinarily require a leap of inventiveness.

30. I also understand that one of ordinary skill in the art must have had a reasonable expectation of success when combining references for claimed subject matter to be obvious.

31. I have been informed and I understand that factors referred to as "objective indicia of non-obviousness" or "secondary considerations" are also to be considered when assessing obviousness when such evidence is available. I understand that these factors can include: (1) commercial success; (2) long-felt but unresolved needs; (3) copying of the invention by others in the field; (4) initial expressions of disbelief by experts in the field; (5) failure of others to solve the problem the claimed subject matter solved; and (6) unexpected results.

32. I also understand that evidence of objective indicia of non-obviousness must be commensurate in scope with the claimed subject matter. I further understand that there must be a relationship, sometimes referred to as a "nexus," between any such secondary indicia and the claimed invention.

33. Finally, I have been informed that one cannot use hindsight to determine that an invention was obvious.

34. I provide my opinions in this declaration based on the guidelines set forth above.

Vervain Ex. 2011 p. 17 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 17
Micron v. Vervain
IPR2021-01550

V. LEVEL OF SKILL IN THE ART

35. I have been informed and understand that the level of ordinary skill in the relevant art at the time of the invention is relevant to inquiries such as the meaning of claim terms, the meaning of disclosures found in the prior art, and the reasons one of ordinary skill in the art may have for combining references.

36. I have been informed and understand that factors that may be considered in determining the level of ordinary skill include: (1) the education of the inventor; (2) the type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) education level of active workers in the relevant field. I have been further informed and understand that a person of ordinary skill in the art is also a person of ordinary creativity.

37. A person of ordinary skill in the art ("POSA") at the time of the alleged invention would have had 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 designing non-volatile memory devices. An individual with an advanced degree in a relevant field would require less experience designing non-volatile memory devices.

38. In view of my educational background (*e.g.*, a Ph.D. in Electrical Engineering obtained in 1989) and decades of experience designing of non-volatile

Vervain Ex. 2011 p. 18 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 18
Micron v. Vervain
IPR2021-01550

memories, as discussed in Section II, I was a person of more than the ordinary level of skill in the art as of July 2011. My opinions herein, however, were formed considering the perspective of an ordinarily skilled artisan.

VI. THE 300 PATENT'S EFFECTIVE FILING DATE

39. I understand that the application leading to the 300 Patent was filed on June 12, 2018.

40. Based on my review of the 300 Patent, I note that it also refers to a July 19, 2011 Provisional Application.

41. For purposes of this declaration, I have been instructed to use July 19, 2011 as the effective filing date of the 300 Patent. My opinions in this declaration were formed from the perspective of a person of ordinary skill in the art as of July 19, 2011, including both the knowledge of a person or ordinary skill in the art at that time as well as how a person of ordinary skill in the art would have understood the prior art.

VII. THE 300 PATENT

A. Technological Background

1. Volatile, Non-volatile, and Flash Memory

42. Computer system memory comes in several varieties. Generally, computer memories store data (e.g., programs and user data) in what is known as "binary" code, *i.e.*, zeros and ones. In one of the simplest forms of memory, a certain number of electrons may be stored on a capacitor to represent a binary 0

Vervain Ex. 2011 p. 19 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 19
Micron v. Vervain
IPR2021-01550

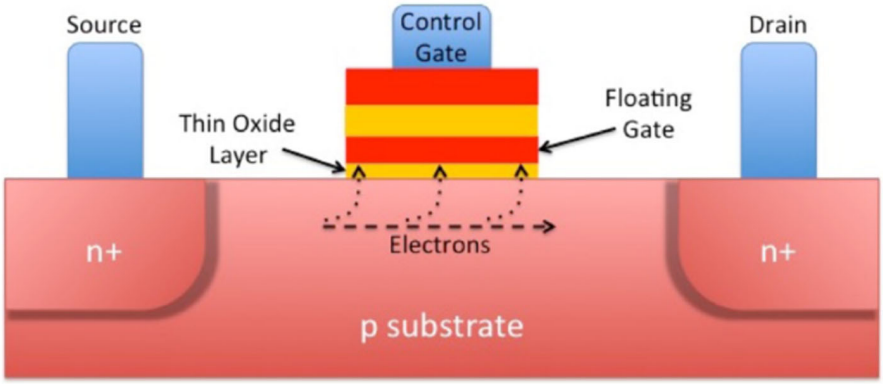
while a different number represents a binary 1. The number of electrons will manifest themselves either as a distinct voltage level or current level that can be obtained from the capacitor.

43. Memory may either be "volatile" or "non-volatile." *Volatile* memory, such as dynamic random access memory (DRAM), needs continuous power to preserve stored data. That is, volatile memory loses its contents when the computer's power is turned off. *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. Unlike volatile memory, non-volatile storage does not require a continuous availability of power in order to retain its data. *See* Ex. 1040, Dipert & Levy at 21.

44. Flash memory has been a known memory type for several decades. It is often found in USB flash drives, MP3 players, digital cameras, digital phones, and more recently, in personal computers and servers. *See* Ex. 1018, Axelson at 17-18, 22-23. One early type of flash memory was known as electrically erasable programmable read only memory ("EEPROM"). Flash has the advantage of being a non-volatile memory, while also being able to be made smaller than traditional non-volatile storage, *e.g.*, disk drives. In addition, unlike disk drives, flash memories do not have mechanical moving parts and thus are more able to withstand sudden movement, making them better suited for portable devices.

Vervain Ex. 2011 p. 20 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 20
Micron v. Vervain
IPR2021-01550

45. As shown below, many flash memory cells use floating gate transistors to store information. Ex. 1014, Prince at 54-55.



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

46. Other flash memory cells use what is known as "charge trap" technology, which is a distinct but similar technology for storing charge in a flash memory cell. Ex. 1019, Micheloni at 115. In both floating gate and charge trap flash memory cells, the amount of charge stored in the cell represents information. For example, a large amount of charge may represent a value of "0," whereas a small amount of charge may represent a value of "1." *Id.* at 20.

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

47. The charge on the floating gate indicates the value that the cell stores by affecting the threshold voltage of the memory cell. The threshold voltage of the memory cell refers to the voltage at which the transistor will begin to conduct

Vervain Ex. 2011 p. 21 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 21
Micron v. Vervain
IPR2021-01550

substantial current across the channel. As described below, the threshold voltage can be set such that the cell stores two values, *i.e.*, either a 0 or 1 (single-level cell, "SLC") or such that the cell stores more than two values (multi-level cell, "MLC"). For example, a multi-level cell may store two bits, representing four values, *i.e.*, 00, 01, 10, or 11.

48. A typical way to program a Flash cell is to apply a voltage pulse to the control gate, and to place the source and drain at ground. Then, electrons will be injected onto the floating gate. This adds charge to the floating gate and, therefore, changes the threshold voltage of the memory cell. Ex. 1019, Micheloni at 24.

49. During programming, the device will periodically perform a verify operation on the cells being programmed. Ex. 1019, Micheloni at 62. After the floating gate has been programmed for a period of time, the device will read the state of the memory cell and attempt to verify that it has been programmed to the proper threshold voltage. If the memory cell is read at the proper state, then programming is complete. If the memory cell does not verify to the proper state, then the programming process is repeated.

50. Different threshold voltages can correspond to different memory states or digital values. For example, if the memory cell has a threshold voltage of 0 volts, this may correspond to a logical high value, whereas a threshold voltage of 1 volt may correspond to a logical low value. The threshold voltage is

Vervain Ex. 2011 p. 22 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 22
Micron v. Vervain
IPR2021-01550

programmable to various threshold values and these memory states can be associated with different logical values. SLC refers to Flash cells that have two states, whereas MLC refers to Flash cells that have more than two states. Ex. 1019, Micheloni at 5-6; Ex. 1015, Varkony at 3:4-7, 7:24-35.

51. Flash memory cells, however, "wear out," *i.e.*, after a certain number of erase and program (write) operations, they become defective and, *e.g.*, remain "stuck" at a 0 or 1 value. *E.g.*, Ex. 1019, Micheloni at 41; Ex. 1016, Friedman at 1:19-37. While in other memories, such as DRAM, memory cells also become defective, the problem is a much bigger issue in flash memories and techniques to deal with this wear have been around, in effect, since the creation of flash memories.

3. Flash Architecture

52. Flash memory is organized into "blocks," "pages," and/or "sectors." Ex. 1019, Micheloni at 27. A block is the smallest erasable unit; an erase operation must occur, at minimum, on an entire block, not a subset of the block. A block is made up of multiple pages. A page is the smallest programmable or writable unit; in a read or write operation, an entire page, not a subset of the page, must be read or written. A page may be made up of one or more sectors. The host (not the flash device) defines the size of the sector. Ex. 1041, Gorobets at 2:57-66.

Vervain Ex. 2011 p. 23 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 23
Micron v. Vervain
IPR2021-01550

53. The size of a block and of a page are often influenced by constraints in the physical design of the memory. For example, blocks in flash memory tend to be bigger than pages because the hardware required to erase cells tends to be bigger than the hardware required to write some number of cells. Thus, 2,048 cells might share the same write hardware (and therefore be in the same page), whereas 128 pages might share the same erase hardware (and therefore be in the same block). In the context of flash memories, a sector is often the same size as a page. Ex. 1019, Micheloni at 40.

54. Flash memories typically comprise many blocks. Because there are multiple blocks, each read and write operation requires both a block address and a page address. An erase operation, on the other hand, requires only a block address, because the erase operation is at a block level. As I discuss below, there are typically two sets of addresses: logical and physical.

55. For flash memories, a page must be erased before it can be programmed (also referred to as written). This is due to how flash memories are programmed. An erase operation, for example, will cause all cells in the block to be set to 1 for an SLC cell and set to 11 for an MLC cell. Taking the case of SLC cells, a program (write) operation, however, can only change bits from 1 to 0 (*i.e.*, in this example, a write operation cannot change a bit from 0 to 1). Thus, a block

Vervain Ex. 2011 p. 24 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 24
Micron v. Vervain
IPR2021-01550

must be erased, setting all cells to 1, before the program operation can correctly write the series of 0s and 1s to the page.

56. Notably, this feature of flash memory is different from many other types of memory. For example, when updating data in DRAM or a hard disk drive, the data may simply be rewritten to the same physical location. The write operation overwrites the existing data at that location. Flash memory is different. Because flash memory program operations can only change bits from 1 to 0 (and not vice versa), a physical location that stores data must be erased before it is written to again.

57. Erasing a block takes a relatively long time, typically an order of magnitude longer than a read or program operation. Ex. 1017, Sinclair at [0008]. As a result, it was known to a POSA that it was undesirable to write a page to a location that was not already erased. *Id.* at [0009]. As discussed, common computer usage involved rewriting data to the physical location in which it was already stored. Yet, under this approach, the device would need to erase a block each time existing data was updated in that particular block—greatly reducing performance. In other words, if the physical location was not already erased, the write operation would require both an erase operation and a write operation.

58. An additional problem is posed by the reality that, in a typical computer system, some files are written to more often than others. Because flash

Vervain Ex. 2011 p. 25 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 25
Micron v. Vervain
IPR2021-01550

memory cells wear out when written to (see above), assigning each file a fixed physical location would result in some physical locations wearing out before others.

59. By 2010, these problems had long been solved. A solution, which was well known to any POSA, was the use of logical addresses (sometimes referred to as virtual addresses) by the host computer, physical addresses to represent physical locations in the flash memory elements, and an address map to translate logical addresses to physical addresses. As explained further below, logical-to-physical address mapping (sometimes referred to as V2P, L2P, and other variation) allowed flash memory devices to (1) immediately write data to an available page without waiting for an erase operation to complete; and (2) spread write/program operations evenly among the device pages, so pages wear at the same rate.

4. Caching

60. 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 at 21-24. In a similar manner, a DRAM cache is used to store data read from hard disk drives. Ex. 1019, Micheloni at 14.

Vervain Ex. 2011 p. 26 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 26
Micron v. Vervain
IPR2021-01550

61. It was well-known to use a variety of caches in flash memory systems. For example, through the well-known process of write caching (sometimes called "write buffering"), a volatile memory cache—typically composed of DRAM or SRAM—is used to store data to be written to the flash device. Ex. 1019, Micheloni at 132 (SRAM cache); *id.* at 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.* at 528-29. Another known benefit is that the volatile memory cache can be written to more quickly. *Id.* at 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.*

62. As the above illustrates, various memory types were used as caches, including, for example, SRAM and DRAM. Ex. 1019, Micheloni at 132 (SRAM cache); *id.* at 44 (DRAM cache).

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

63. The location of data in memory is identified by a number known as an "address." Conventionally, each memory address corresponds to one byte (8 bits) of data. This scheme is known as byte addressing. In a byte-addressed system, a page that stores 512 bytes would have 512 addresses within that page to identify

Vervain Ex. 2011 p. 27 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 27
Micron v. Vervain
IPR2021-01550

specific bytes of data. Similarly, a block that has 128 pages would have 512 times 128 or 65,536 individual addresses within that block. Conventionally, these addresses are contiguous, and each page and block would be associated with a range of addresses. For example, addresses 0 through 511 might correspond to page 0, addresses 512 to 1023 might correspond to page 1, etc. Addresses 0 through 65,535 might correspond to block 0. More generally, a unit of data storing n bytes would conventionally have n addresses within that unit. Taken together, a set of addresses is known as an "address space" or a "memory space." *See, e.g.*, Ex. 1051, Microsoft Computer Dictionary at 20 (defining "address space" as "[t]he total range of memory locations addressable by a computer"); Ex. 1052, Radke Appl.[1] at [0037] (characterizing the combination of SLC physical addresses as "the SLC . . . memory space").

64. By 2010, a POSA would have known about the use of logical-to-physical address mapping to solve many well-known issues with flash memory. In fact, by 2010, for almost all flash memory implementations, using logical-to-physical mapping would have been an assumption of the design.

65. The logical address space is the set of addresses that the higher-level system, *e.g.*, a host, uses. Thus, for example, operating systems (e.g., Microsoft

[1] US 2009/0300269.

Vervain Ex. 2011 p. 28 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 28
Micron v. Vervain
IPR2021-01550

Windows) or software (e.g., Microsoft Word) would use logical addresses to identify locations within a storage device to read and/or write. A logical address is translated ("address mapped") into a physical address before access to the memory occurs. The flash device uses physical addresses, *i.e.*, the addresses that specifically identify blocks, pages, and/or sectors in the flash memory. For example, the host would direct read and write operations to logical addresses. An address translation scheme translates the logical addresses into physical addresses. Specifically, a table (an "address map") maintained by the flash memory (or its controller) is used to translate logical addresses to physical addresses. Ex. 1018, Axelson at 29-30. The system for mapping logical addresses to physical addresses is sometimes called a "flash translation layer" or "FTL." Ex. 1019, Micheloni at 492.

66. For explanation purposes, a simplified[2] translation table with two entries is shown below:

Logical Address	Physical Address
page 3	block 10, page 6

[2] A logical address may be expressed as only a page address (often referred to as an LPA), block address (LBA), or other unit, which is translated into a physical page and block address.

Vervain Ex. 2011 p. 29 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 29
Micron v. Vervain
IPR2021-01550

page 4	block 12, page 2

67. The table and subsequent examples use blocks and pages as a shorthand to identify logical and physical address ranges; each of these blocks and pages would correspond to a particular range of addresses. For example, in a byte-addressed system with 512-byte pages and 128-page blocks, logical page 3 might correspond to logical addresses 1536–2047 (*i.e.*, addresses 3×512 to $4 \times 512 - 1$), and physical block 10, page 6 might correspond to physical addresses 658432–658943 (*i.e.*, $10 \times 65536 + 6 \times 512$ to $10 \times 65536 + 7 \times 512 - 1$).

68. When the host directs a read operation to page 3, the flash device (or its controller) would translate this logical address to physical block 10, page 6, which is the actual address of the data.

69. The use of logical-to-physical address mapping was textbook, well understood knowledge to any POSA. For example, the Micheloni textbook provides the following figure showing "[l]ogical to physical block management":

Vervain Ex. 2011 p. 30 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 30
Micron v. Vervain
IPR2021-01550

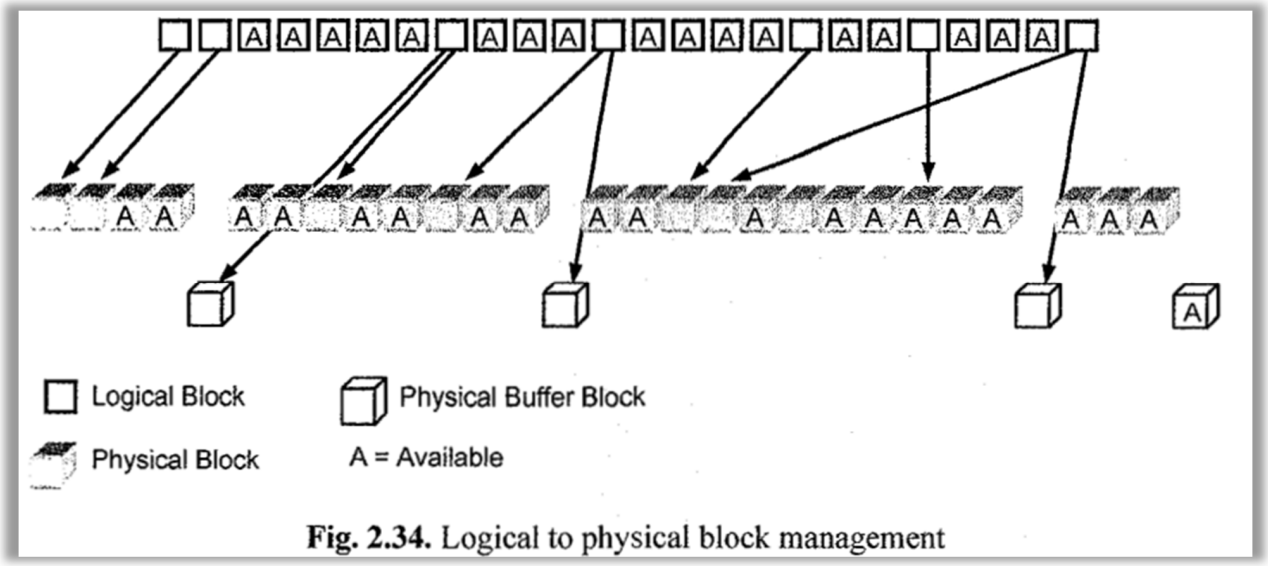


Fig. 2.34. Logical to physical block management

Ex. 1019, Micheloni at 41.

70. A logical-to-physical addressing scheme has several benefits. For example, logical to physical address mapping simplifies the host operation in the case of defective memory cells. Specifically, certain cells (at physical addresses) may become defective. In this scenario, the host cannot successfully direct operations at these physical addresses because the cells are defective. But rather than the host keeping track of these defective cells, the controller of the Flash memory can instead keep track. Ex. 1042, Conley at 6:16-21 (discussing defective cell and block remapping as prior art). Thus, the host device need not concern itself with whether the cells are defective and can simply provide the controller the same logical address as if the cells never became defective. Let us assume that the host writes to logical address page 3, which the Flash device translates to physical address block 10, as shown below:

Vervain Ex. 2011 p. 31 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 31
Micron v. Vervain
IPR2021-01550

Logical Address	Physical Address
page 3	block 10, page 6

71. Now, let's assume that block 10, page 6 become defective (likely, the device would deem the whole block 10 defective). In this case, the Flash device can substitute a different physical block, as shown below:

Logical Address	Physical Address
page 3	block 18, page 2

72. In this case, to the host, the addressing does not change, and it need not concern itself with the internal working of the Flash device. As can be seen from the above, the host system can still use the same logical address of page 3. In this way, the Flash device's use of logical to physical address mapping simplifies the host's use of the flash memory.

73. A similar bad-block-remapping technique is shown in the figure below:

Vervain Ex. 2011 p. 32 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 32
Micron v. Vervain
IPR2021-01550

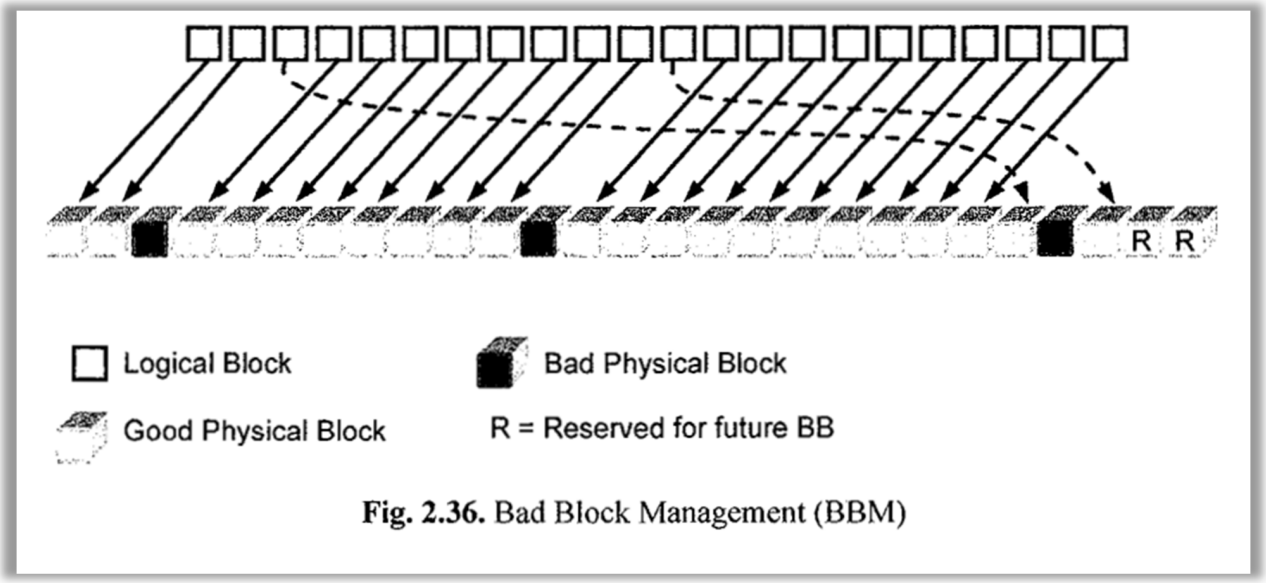


Fig. 2.36. Bad Block Management (BBM)

Ex. 1019, Micheloni at 42.

74. A logical-to-physical address scheme similarly enables wear leveling techniques. One well-understood wear leveling technique is known as "dynamic wear leveling." In dynamic wear leveling, when a flash memory device writes to a logical address, instead of erasing the associated physical address and writing the new data to that same physical address, the flash device instead selects an already-erased physical address and writes the new data to this physical address. The selected physical address is the least worn available physical address (*i.e.*, the physical address that had been written to the fewest number of times). The address map (also sometimes called an address translation table) is then updated to map the logical address to the selected physical address. Ex. 1043, You at 1:46-48. For example, with reference to the above table, assume that a host directs a write operation to logical address page 3. Rather than writing the data to physical block

Vervain Ex. 2011 p. 33 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 33
Micron v. Vervain
IPR2021-01550

10, page 6 (which is associated in the table with page 3) by erasing it and then programming it, the data is instead written to an already-erased page that has relatively little wear. If block 14, page 6 is an already-erased block that has relatively little wear, it can be selected for this operation, and the flash device writes the data to this block. The table would then be updated as shown below:

Logical Address	Physical Address
page 3	block 14, page 6
page 4	block 12, page 2

75. Because the flash device can choose any already-erased page/block to write the data to, this technique provides the device the opportunity to select pages/blocks that have been written to relatively few times and thus balance the overall number of writes throughout the device. More specifically, it was known to maintain a list of empty (*i.e.*, available) blocks ordered by the number of previous write/erase cycles on the respective blocks. Then, when new data is to be written, it can be "programmed into a free block with the fewest write/erase cycles." Ex. 1019, Micheloni at 494; *see also* 300 Patent at 3:28-30. In other words, on each incoming write operation, the flash controller can write the data to the next available block, which is the free block with the fewest write/erase cycles. Ex. 1029, Kolokowsky at 2:53-3:13 (including pointers to the next available range of addresses); Ex. 1028, Paley at [0201]-[0202] (disclosing "free block

Vervain Ex. 2011 p. 34 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 34
Micron v. Vervain
IPR2021-01550

management" to provide next available locations for writes). The 300 Patent's background section acknowledges that this technique was known. 300 Patent at 3:28-30. This technique is sometimes known as "dynamic wear leveling," *id.*, or "first level wear leveling." Ex. 1019, Micheloni at 494. A POSA would have known multiple techniques for performing dynamic wear leveling. *See id.*

76. In another well-known wear leveling technique, if a block with data already stored in it (*i.e.*, a first block) has a write/erase account below a threshold value, the data in that block will be moved to a second block. Then, the next data to be written will be written to the first block, which is now available. The 300 Patent's background section acknowledges that this technique was known. 300 Patent at 3:31-35. The 300 Patent calls this technique "static wear leveling," *id.*, but it is also sometimes called "second level wear leveling." Ex. 1019, Micheloni at 494.

77. The prior art wear-leveling techniques have several well-understood advantages. By selecting an already-erased block for each write operation, the flash device can immediately program the data without initiating and waiting for completion of an erase operation. In addition, by spreading the writes throughout the device's physical addresses, the device will wear more evenly. *See* Ex. 1018, Axelson at 26, 29-30. If the device wears unevenly, blocks will fail unnecessarily early, reducing the overall memory capacity of the device and potentially leading

Vervain Ex. 2011 p. 35 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 35
Micron v. Vervain
IPR2021-01550

to system incompatibility because the host may rely on a capacity that is larger than the reduced capacity. *See* Ex. 1043, You at 2:41-3:3.

78. Logical-to-physical address mapping and the techniques it enables (e.g., wear leveling and bad block management) were well understood by 2010, as is evidenced by their description in an introductory textbook on NAND flash memories. Ex. 1019, Micheloni at 41, 492.

6. Flash Translation Layer ("FTL")

79. By 2010, a POSA would have understood what a flash translation layer is; indeed, at that time, virtually any flash memory system would include a flash translation layer. A POSA would have understood that 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 locations). At its most basic, this involves translating logical to physical addresses. But more feature-rich FTLs also performed memory management operations such as wear leveling and bad block management. Take, for example, this diagram from the Micheloni textbook showing the FTL's place in an SSD communicating with an operating system ("OS"):

Vervain Ex. 2011 p. 36 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 36
Micron v. Vervain
IPR2021-01550

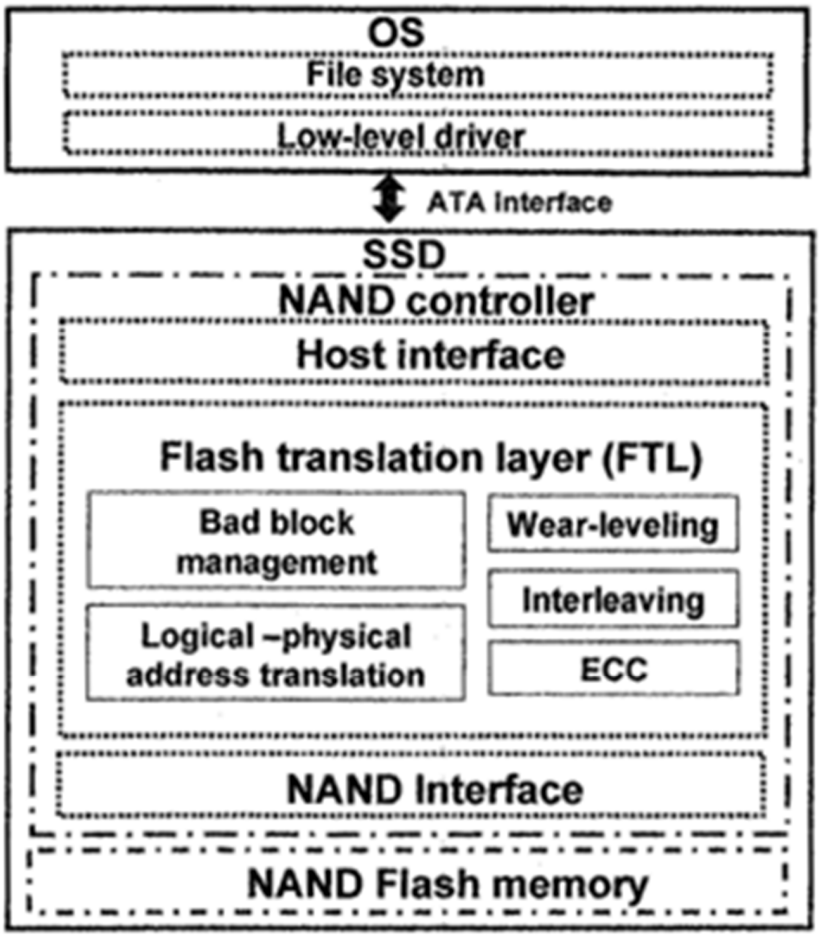


Fig. 18.3. Software architecture of SSD

Ex. 1019, Micheloni at 517. The FTL shown above performs bad block management, wear leveling, logical-to-physical address translation, and other functions.

80. A POSA would have understood that an FTL is related to, but is a distinct concept from, a controller. In short, the FTL is the function and the controller is the structure that often implements that function. Ex. 1019, Micheloni at 492 ("Usually it is the Controller . . . taking care of the FTL."). But a POSA

Vervain Ex. 2011 p. 37 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 37
Micron v. Vervain
IPR2021-01550

would recognize that FTLs are not necessarily implemented in a controller. For example, the host itself (e.g., the operating system) could potentially implement the FTL. *Id.* at 10; Ex. 1010, Dusija at [0062].

81. Thus, compare/contrast Micheloni's "software architecture of SSD," above, with Micheloni's "hardware architecture of SSD," below. Ex. 1019, Micheloni at 516. Notice that there is no hardware element called the FTL. That functionality is implemented by the "NAND controller."

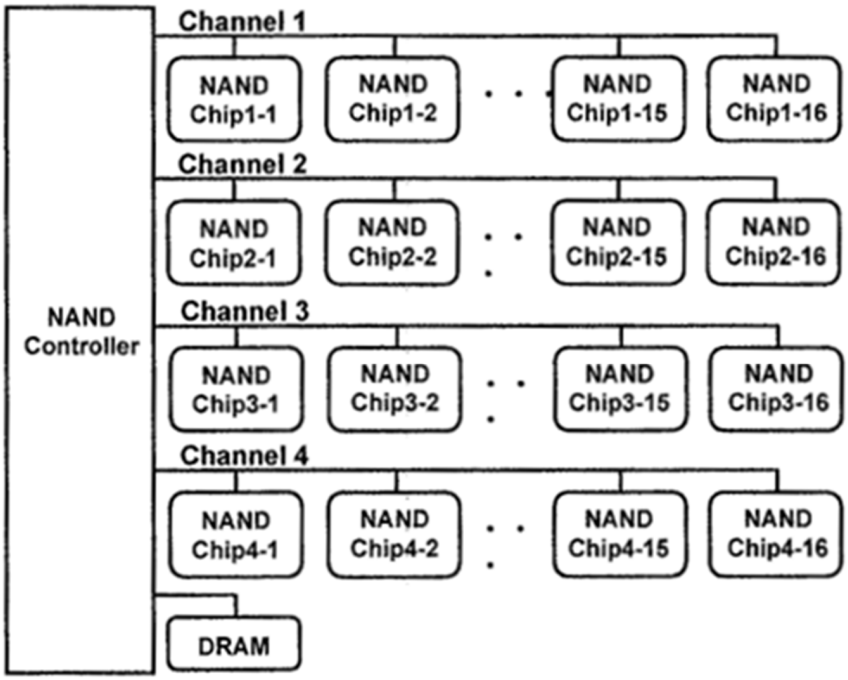


Fig. 18.2. Hardware architecture of SSD

82. Indeed, a POSA would have recognized that the FTL can be implemented as part of a NAND controller. Ex. 1019, Micheloni at 516-17 ("Figure 18.3 shows the software architecture of SSD. The NAND controller is composed of the host interface, the Flash translation layer (FTL) and the NAND

Vervain Ex. 2011 p. 38 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 38
Micron v. Vervain
IPR2021-01550

interface. FTL is the central ***part of the NAND controller*** and perform a bad block management, a logical-physical address translation, a wear-leveling, ECC and an interleaving. The intelligent write algorithm described in Sects. 18.5 and 18.6 is also controlled with FTL.") (emphasis added).

83. The 300 Patent's background section uses the term "FTL" in a manner consistent with the above. The 300 Patent's background section states that "FTL is the driver that works in conjunction with an existing operating system (or, in some embedded applications, as the operating system) to make linear flash memory appear to the system like a disk drive, *i.e.*, it emulates a HDD." 300 Patent at 3:3-7. This is logical-to-physical address mapping. The "FTL further manages the flash so that there are clean/erased places to store data." *Id.* at 3:11-12. This can include bad block management, garbage collection, and wear leveling. And the patents acknowledge that the FTL is implemented "as firmware." *Id.* at 3:14-16. This reflects the typical case in which the FTL is implemented as part of a controller, which typically runs firmware.

84. In sum, a POSA would have recognized a flash translation layer to be a description of functionality that could be implemented in different places, but most typically in a flash memory controller.

Vervain Ex. 2011 p. 39 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 39
Micron v. Vervain
IPR2021-01550

7. Speed and Durability Considerations for MLC and SLC Cells

85. A POSA understood that MLC and SLC have distinct advantages and disadvantages. A POSA would have understood that MLC stores data more densely and cheaply than SLC. A POSA would have also understood, however, that the write speed for MLC is markedly slower than for SLC. Ex. 1020, Y. Lee at [0005]. Thus, it was well known that the system will operate faster if frequently updated data is written to SLC instead of MLC. *Id.* In addition, it was well-known that SLC was more durable (*i.e.*, could sustain more write operations without degrading) than MLC. 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.* at [0006]; Ex. 1019, Micheloni at 98. 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. *See* Ex. 1020, Y. Lee at [0005].

86. 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. *See, e.g.*, Ex. 1021, Kim at 3:45-58; Ex. 1010, Dusija at [0018]; Ex. 1022, Gavens at [0018]; Ex. 1023, C. Lee at 1:32-42, 1:53-67.

Vervain Ex. 2011 p. 40 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 40
Micron v. Vervain
IPR2021-01550

8. Data Integrity Tests

87. Data integrity tests, broadly speaking, can include writing data to memory and, after the write operation has been completed, performing a subsequent operation (usually a read operation) to ensure that the data was written without errors. *See* Ex. 1022, Gavens at [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 at 10:41-53; Ex. 1025, Microsoft Computer Dictionary at 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").

B. Summary of the 300 Patent's Disclosure

88. 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 at 4:17-51.

89. 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 at 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*. at 4:30-32. The data to be written to NAND flash is initially stored in a bank of DRAM. *Id.* at 5:59-61,

Vervain Ex. 2011 p. 41 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 41
Micron v. Vervain
IPR2021-01550

6:17-20. The device controller then reads the data and stores the data into its memory within the device controller. *Id.* at 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.* at 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.* at 6:30-32. The controller compares the newly stored data to the maintained (or retained) data. *Id.* at 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.* at 6:40-52.

90. 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. Dusija squarely discloses this type of data integrity test. *See* Section IX.B.1.k (limitation [1.H]).

Vervain Ex. 2011 p. 42 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 42
Micron v. Vervain
IPR2021-01550

91. 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 at 6:53-59.[3] The specification discloses:

> This could be accomplished within the device controller 14 described above, which could simply maintain a count of those blocks that are accessed (written to) most frequently, and, on a periodic basis, such as, for example, every 1000 writes, or every 10,000 writes, transfer the contents of those blocks into the SLC NAND flash memory module 28.

Id. at 6:59-65.

92. 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. Specifically, Sutardja discloses comparing write counts with a threshold value and, if the write count is greater than that threshold value (*i.e.*, after a predetermined number of writes), transferring the contents of an MLC block to an SLC block independent of any host write operation, *i.e.*, essentially as part of a background house-keeping data management process. *See* Section IX.C.1

[3] While the specification also discloses allocating "cold blocks" (*i.e.*, those that receive infrequent writes) into MLC memory, claim 10 only requires allocating hot blocks from MLC to SLC.

Vervain Ex. 2011 p. 43 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 43
Micron v. Vervain
IPR2021-01550

(Sutardja). In this way, the MLC blocks that are most frequently written to (thus surpassing the write threshold) will be transferred to SLC blocks. *Id.*

C. The 300 Patent's Prosecution History

93. I have reviewed the file history. I understand that claim 1 of the 300 Patent, as initially filed, 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, 300 File History at 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*. at 55, 59. The Examiner again rejected the claims as obvious. *Id.* at 73. The Applicant then added an additional limitation reciting that a data integrity test results in data being transferred (in addition to being remapped, as was already recited). *Id*. at 91-94. The Examiner continued to find the claims to be obvious. *Id.* at 145-47.

94. 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." 300 File History at 173. The controller then reads back the data stored in MLC memory (*i.e.*, the "stored data"). *Id.* at 180. The controller performs the data integrity test

Vervain Ex. 2011 p. 44 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 44
Micron v. Vervain
IPR2021-01550

"by comparing the stored data to the retained data in the random access volatile memory." *Id.* at 173. The Examiner allowed the application after this amendment. *Id.* at 173, 180, 204.

95. In issuing the notice of allowance, the Examiner pointed to two pages of the Applicant's response to the prior Office Action. 300 File History at 210 (citing *id.* at 179-80). 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 [sic] 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.

Vervain Ex. 2011 p. 45 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 45
Micron v. Vervain
IPR2021-01550

Id. at 180.[4]

96. In the above statement, the Applicant repeatedly referenced a "nonvolatile memory." A POSA would understand this reference to be a typographical error, and a POSA would have understood that the Applicant meant to say "volatile memory" in each of these instances. Recall that the independent claims recite a "random access volatile memory," and MLC and SLC "nonvolatile memory element[s]." The independent claims further recite that, in a "Write access operation," the controller "retain[s] [the] stored data in the random access volatile memory." The claims further recite that, after the write operation, the controller "compar[es] the stored data to the retained data in the random access volatile memory."

97. A POSA would have recognized that it is the volatile memory—not a non-volatile memory, that would store both the "retained data" and the "stored data." First, the claims expressly state that the retained data is kept in volatile

[4] In this office action response, Applicant distinguishes "nonvolatile memory" from cache and refers to "nonvolatile memory" performing actions (*i.e.*, retaining data) that the 300 Patent and its file history only refer to as being performed by the volatile memory. It is assumed, for purposes of this declaration, that Applicant intended to refer to "volatile memory" for this discussion.

Vervain Ex. 2011 p. 46 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 46
Micron v. Vervain
IPR2021-01550

memory. Thus, when Applicant suggested that "that entire block has to be retained in one portion of the [non-volatile] memory," a POSA would have recognized that as a mistake. Second, were the "retained data" stored to non-volatile memory, it wouldn't be "retained" because it would be stored in the same memory as the "stored data." Third, it makes little sense to store the retained data to flash memory because the whole point of the data integrity test is to detect errors caused by writing data to non-volatile memory; this purpose would be frustrated by writing *both* the stored data and retained data to non-volatile memory. Fourth, it was well-known to a POSA that a controller's working memory would be a volatile memory; flash memory is too slow for that purpose. Ex. 1019, Micheloni at 144. Thus, the recited "comparing" step would necessarily be performed while the data was in volatile memory.

98. Thus, a POSA would have understood Applicant's statements above to distinguish 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).

99. 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

Vervain Ex. 2011 p. 47 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 47
Micron v. Vervain
IPR2021-01550

data" (*i.e.*, the retained data) and a data integrity test using a "comparison" with retained data. 300 File History at 180. As I discuss below in the claim construction section, 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." *See* Section VIII.B.

VIII. CLAIM CONSTRUCTION

100. I am not a legal expert or an attorney and offer no opinions on the law. I understand that claim construction is a matter of law. However, I have been informed by counsel of the legal standards that apply to claim construction in an *inter partes* review, and I have applied them in forming my opinions. I understand that during an *inter partes* review, the same claim construction standard used in federal district court is applied when addressing the meaning of claim terms.

101. I have been informed that the words of a claim are generally given the ordinary and customary meaning that the term or phrase would have to a person of ordinary skill in the art at the time of the invention in view of the surrounding claim language, the specification, and the file history (collectively, the "intrinsic evidence" or "intrinsic record").

Vervain Ex. 2011 p. 48 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 48
Micron v. Vervain
IPR2021-01550

102. I also understand that courts may consider extrinsic evidence, such as expert and inventor testimony, dictionaries, and learned treatises (collectively, the "extrinsic evidence" or the "extrinsic record"), but that such extrinsic evidence should be given less weight than the intrinsic evidence.

103. I understand a means-plus-function limitation is construed to be the specific structure disclosed in the patent that corresponds to the claimed function. I also understand that construing a means-plus-function claim term is a two-step process. First, one identifies the claimed function. Second, one determines what structure, if any, disclosed in the specification corresponds to the claimed function.

104. I have applied the claim constructions below in rendering the opinions set forth in this declaration.

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

105. Claims 1 and 12 recite "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 . . . ." 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 . . . ." My understanding is that Micron does not contend that an express construction of "data integrity test" is necessary. To the extent that the Board determines that a construction is necessary, in the context of

Vervain Ex. 2011 p. 49 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 49
Micron v. Vervain
IPR2021-01550

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."

106. In the Summary of the Disclosure section, the 300 Patent 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 at 4:30-32; *see also id.* at 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").

107. 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 integrity test "after at least a Write access operation," not after all write access operations. 300 Patent at 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)

108. 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).

Vervain Ex. 2011 p. 50 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 50
Micron v. Vervain
IPR2021-01550

109. 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."

110. A POSA would have reached this conclusion based on 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.

111. During prosecution, the claims—which at the time recited a data integrity test without specifying *how* it was performed—were rejected as obvious. 300 File History at 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.* at 173. The Applicant further amended the claimed "data integrity test" to be performed "after . . . at least

Vervain Ex. 2011 p. 51 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 51
Micron v. Vervain
IPR2021-01550

a Write access operation . . . by ***comparing the stored data to the retained data in the random access volatile memory***." *Id.* (emphasis added).

112. 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." 300 File History at 180; *see id.* at 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 at 6:15-38 (describing the comparison as occurring in "memory within the device controller"). A POSA would understand, based on this description and the claim language, that 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." 300 File History at 103, 180, 202.

Vervain Ex. 2011 p. 52 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 52
Micron v. Vervain
IPR2021-01550

113. Further, a POSA would have understood the Applicant to have 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. 300 File History at 189. The Examiner found the Applicant's distinctions persuasive in allowing the claims. *Id.* at 189, 210. Thus, in view of the prosecution history, a POSA would understand that the claims' "stored data" and "retained data" are actual copies of the originally written data, not representations or abstractions of that data, *e.g.*, ECC bits.

114. This understanding would be confirmed by the specification, which only describes a comparison with stored and retained data that are actual copies of the original written data. 300 Patent at 6:17-52.

C. "periodically"

115. 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."

116. A POSA would not have understood the 300 Patent to disclaim or limit the term "periodically" from its ordinary meaning. Instead, a POSA would have understood the 300 Patent to give examples of transfers occurring "on a periodic basis," that is, "every 1000 writes, or every 10,000 writes." 300 Patent at 6:62-63. Thus, "periodically" should have its plain and ordinary meaning, which

Vervain Ex. 2011 p. 53 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 53
Micron v. Vervain
IPR2021-01550

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 at 921.

D. Other Terms

117. I have applied the plain and ordinary meaning of all other terms.

IX. HOW THE CHALLENGED CLAIMS ARE UNPATENTABLE

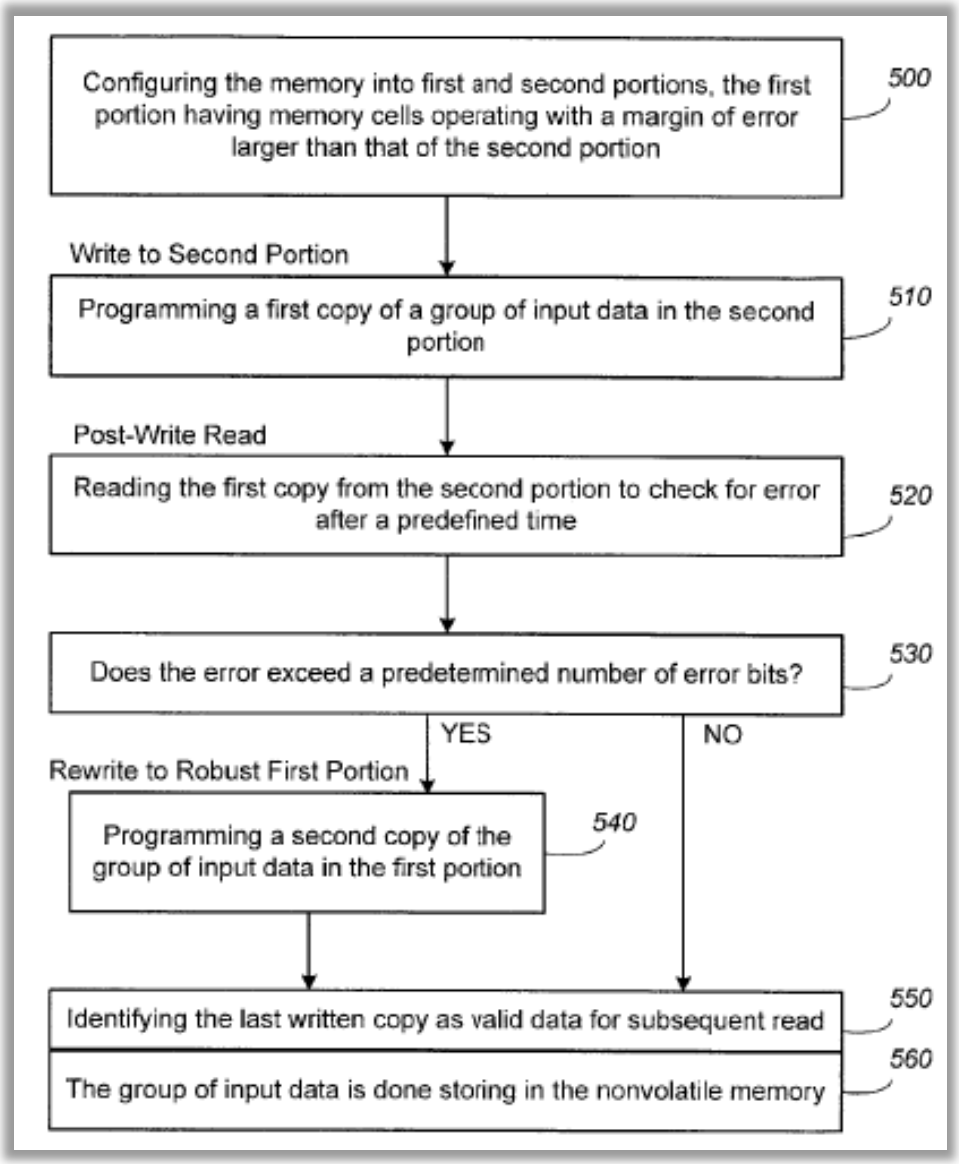
A. Prior Art Overview

1. Dusija

118. 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 at Abstract, [0019] (preferred embodiment's "first portion" is SLC and "second portion" is MLC), [0059]. As I describe in the background section, flash memories with both SLC and MLC were well known at the time.

119. Dusija discloses a data integrity test that rewrites data if an original write to MLC fails. Specifically, Dusija refers to this data integrity test as a "post-write read" method of checking errors, performed after data is written into the memory array. Dusija at [0111], [0119], Fig. 15. The "post-write read" process is depicted in Figure 15 below:

Vervain Ex. 2011 p. 54 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 54
Micron v. Vervain
IPR2021-01550



Id. at Fig. 15. In this method, data is written in the second portion of memory (*i.e.*, MLC) (step 510). *Id*. The aim here is to try to write the data initially to a more dense storage portion. 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*. at [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

Vervain Ex. 2011 p. 55 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 55
Micron v. Vervain
IPR2021-01550

incorporation of Paley, a POSA would have understood that Dusija's cache could be a volatile random access memory. *Id.* at [0061]; Limitation [1.A]. 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). Dusija at [0113], [0116]. In other words, if the denser storage option fails, Dusija attempts to write the data to the more robust storage area that is less error prone. 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*. at [0117]. As a result, subsequent host accesses will be directed to the copy in SLC (step 550). *Id*.

Vervain Ex. 2011 p. 56 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 56
Micron v. Vervain
IPR2021-01550

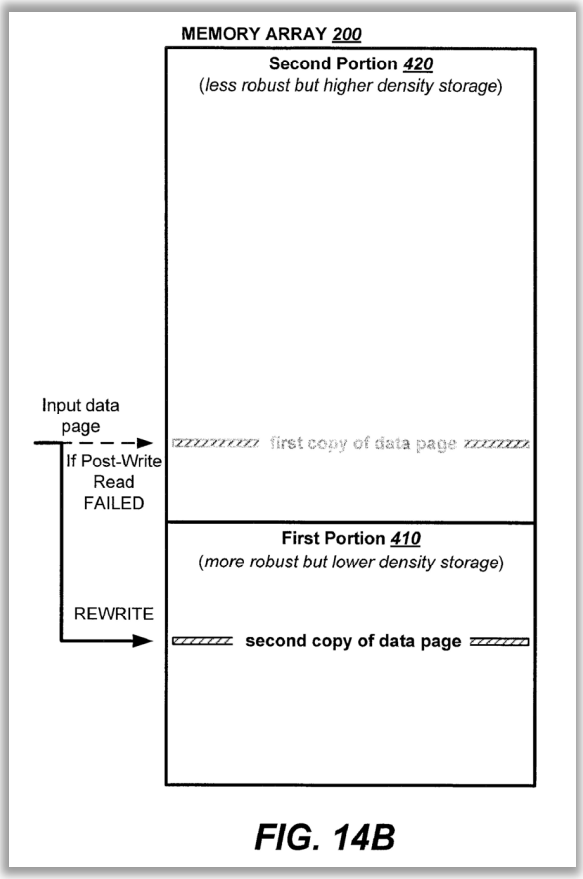


FIG. 14B

Id. at Fig. 14B.

2. Sutardja

120. Sutardja relates to "hybrid non-volatile solid-state memories" (Ex. 1011, "Sutardja," at [0002]) that has a "first solid-state nonvolatile memory" (which may be MLC) and a "second solid-state nonvolatile memory" (which may be SLC). *Id*. at [0108]. As I noted in the background section, such "hybrid" drives including both MLC and SLC were well-known at the time. *See* Section VII.A.7.

121. Sutardja discloses a hot blocks feature that moves data from MLC to SLC memory. Sutardja discloses "periodically" performing a "data shift analysis"

Vervain Ex. 2011 p. 57 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 57
Micron v. Vervain
IPR2021-01550

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 at [0148], [0149], [0167].

122. A POSA would have recognized the purpose of Sutardja's hot blocks feature. It was well-known to those familiar with memory design that some logical addresses (*i.e.*, logical blocks) are written much more frequently than others. For example, some operating system files—such as the bootloader—might only ever be written once, when the operating system is installed. Yet other files—such as an operating system's page file—might be written to every second that a computer is on. User files, such as text documents, typically have write frequencies in between these kinds of files. A POSA would have recognized that blocks that have, in the past, been frequently updated are likely to be frequently updated in the future. Thus, a POSA would have recognized the benefit of moving blocks that have been frequently updated to SLC. This would allow these blocks to be written to more quickly (as writes to SLC tend to be faster than writes to MLC), and also improve durability (as SLC wears out a less as a result of write operations than MLC).

Vervain Ex. 2011 p. 58 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 58
Micron v. Vervain
IPR2021-01550

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

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

1. Claim 1

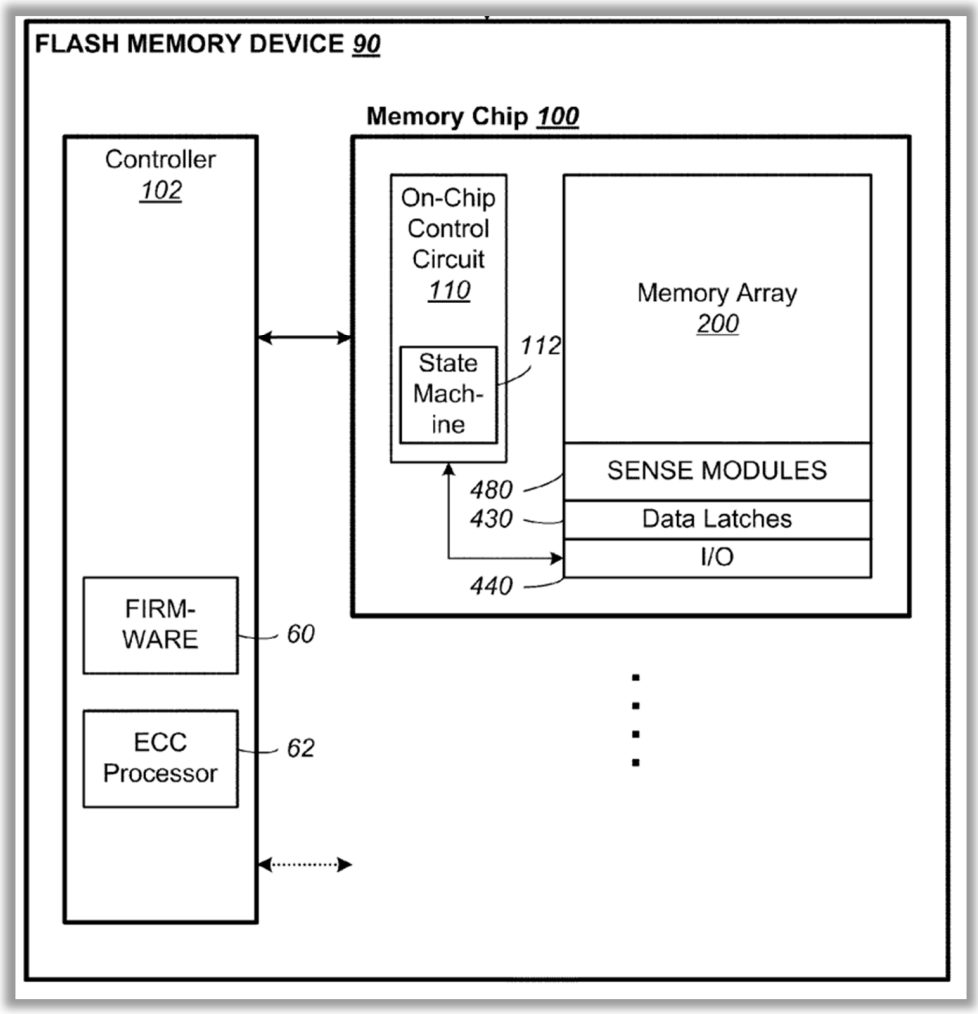
a. [1.PRE] "A system for storing data comprising:"

124. I understand the question of whether a preamble is limiting is a question of claim construction. I do not provide an opinion as to whether the preamble of claim 1 is limiting.

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

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

Vervain Ex. 2011 p. 59 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 59
Micron v. Vervain
IPR2021-01550



Dusija at Fig. 1, [0059].

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

126. Dusija discloses this limitation.

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

Vervain Ex. 2011 p. 60 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 60
Micron v. Vervain
IPR2021-01550

128. Dusija also discloses, or at least renders obvious, "*memory space containing volatile memory space*" for several reasons.

129. 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 data written to and subsequently read from nonvolatile memory. Dusija at [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 Computer Dictionary at 497, 558. Further, during the prosecution of the 300 Patent, the

[6] Dusija states that its "cache" could be, "[i]n an alternative embodiment," flash memory (what Dusija calls the "first portion"). Dusija at [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. That the cache is preferably a volatile memory would be intuitive to a POSA. The benefits of a volatile memory for caching were well understood. For example, a volatile memory tends to be faster than flash memory and also has superior write endurance. Therefore, a POSA would have recognized that it would be obvious to implement Dusija's cache using volatile memory.

Vervain Ex. 2011 p. 61 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 61
Micron v. Vervain
IPR2021-01550

Examiner repeatedly took official notice that SRAM and DRAM are "volatile memory types" that are often used as "cache memory." 300 File History at 48, 76. The Applicant did not contest this use of official notice.

130. Second, Dusija discloses controller 102, which includes firmware 60, that manages memory operations. Dusija at Fig. 1, [0059], [0062]. A POSA would have understood, and at least found it obvious, that the controller includes RAM in order to buffer data and commands, execute the firmware, and manage the memory operations. Ex. 1019, Micheloni at 132 (showing "Data SRAM" as being part of a flash device's control logic). In simple terms, the RAM can be thought of a scratch-pad that the controller uses to perform its function.

131. Third, Dusija incorporates Paley, which also discloses this limitation. Dusija at [0061].

132. It is my understanding that there is a particular legal standard for when a patent publication is treated by the Board as including the contents of another publication to which it refers. I do not opine as to whether that standard is met by Dusija's reference to Paley. However, it is my opinion that a POSA reading Dusija's disclosure would have understood it to include Paley's disclosure in its entirety, and a POSA would have retrieved and reviewed Paley's disclosure. A POSA would have been motivated to do so at least because Dusija refers to Paley as disclosing "[a] preferred block management system." Dusija at [0061]. A

Vervain Ex. 2011 p. 62 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 62
Micron v. Vervain
IPR2021-01550

block management system of some form is necessary to implement Dusija's teachings, so a POSA would have retrieved Paley to retrieve further information on how to implement the block management system. A POSA would have been able to retrieve Paley's disclosure because Dusija provides the application number of Paley, and a POSA would be able to use this application number to retrieve Paley's disclosure. For example, a POSA had available websites such as Public PAIR, Google Patents, or Free Patents Online. A POSA also would have understood that when a patent such as Dusija specifically cites such a reference in this fashion, its teachings are likely applicable or provide additional useful detail. Having retrieved Paley, a POSA would have read and considered its entire disclosure at least because: (1) Dusija incorporates "the entire disclosure" of Paley by reference, Dusija at [0061]; and (2) with Paley in hand, a POSA would want to understand the reference in its entirety.

133. Paley discloses a controller for a flash memory device that includes RAM ("*volatile memory space*"). Ex. 1028, Paley at 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*. at [0197] (emphasis added). Thus, Paley discloses "*memory space containing volatile memory space and nonvolatile memory space*" and that it was customary for RAM to be employed as cache in a flash memory device's controller. *Id*. A POSA would have understood

Vervain Ex. 2011 p. 63 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 63
Micron v. Vervain
IPR2021-01550

Dusija's incorporation of Paley to disclose "*memory space containing volatile memory space and nonvolatile memory space*."

134. Fourth, Paley fully incorporates the disclosure of U.S. Patent No. 5,936,971 to Harari et al. (Ex. 1049), explaining that Harari discloses "[u]sing RAM in a write cache operating with a flash memory." Ex. 1028, Paley at [0027].

135. Again, I do not opine on whether the legal standard for incorporation by reference is met with respect to Paley and Harari. However, as with Dusija's reference to Paley, a POSA would have understood Paley (and, by consequence, Dusija) to include Harari's disclosure. This is at least because Paley explains that "[a]ny and all patents . . . referenced herein are hereby incorporated herein by those references in their entirety for all purposes." Ex. 1028, Paley at [0004]. Further, 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." Thus, Dusija incorporates by reference "the entire disclosure of [Paley]." 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]. Thus, Dusija specifically incorporates Paley's block management system that operates Paley's cache. Similarly, Paley incorporates Harari, referencing Harari's disclosure of

Vervain Ex. 2011 p. 64 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 64
Micron v. Vervain
IPR2021-01550

"[u]sing RAM in a write cache operating with a flash memory." Id., [0027]. And even if a POSA would not have understood Paley to incorporate Harari in a formal sense, a POSA would have retrieved Harari based on the reference to Harari in Paley's disclosure. This is because a POSA would have recognized the benefits of write caching and would have sought to understand the specific write caching approach recommended by Paley.

136. Harari's system includes "a cache system 701 as part of the controller" for temporarily holding files prior to writing them to flash ("*nonvolatile memory*"). Ex. 1049, Harari at 12:20-35, 13:24-42, 13:58-14:11. Harari further demonstrates the ordinary knowledge of a POSA and that a POSA would have understood that Dusija's disclosure of a controller and of caching data to both disclose (and at least render obvious) a "*memory space containing volatile memory space*" operating in a flash memory system ("*nonvolatile memory space*").

137. Dusija's cache is not a "conventional cache" as a POSA would have understood the Applicant to have used that term. As Section IX.B.1.k (limitation 1.H) illustrates, Dusija's cache is not a "conventional cache" because it can store ***both*** the retained written data (*i.e.*, "*retained data*") and the read back data (*i.e.*, "*stored data*"). Moreover, it was well known for caches to be large enough to do so. For example, Harari's cache is capable of storing multiple files. Ex. 1049, Harari at 13:29-30. More generally, as I explained above in the background

Vervain Ex. 2011 p. 65 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 65
Micron v. Vervain
IPR2021-01550

section, it was well-known to use volatile memory to cache data read from and written into the flash memory. *See* Section VII.A.4.

138. It certainly would have been obvious to use volatile RAM in Dusija's system. For example, it would have been obvious to use RAM in conjunction with Dusija's controller, and to implement Dusija's cache as RAM. First, with respect to the hardware specifics, Dusija generically discloses a "cache." Thus, a POSA would have been motivated to consider teachings incorporated into Dusija itself (including Paley and Harari), and the ordinary knowledge of a POSA. Second, as I explained in the background section, using a RAM as a data cache was well known and typical. *See* Section VII.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 its relatively high speed, strong write endurance, and random access capability. Ex. 1019, Micheloni at 516 (showing "DRAM" as being part of a "NAND Controller"). And implementing a cache using RAM was routine and highly predictable. *See* Section VII.A.4.

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

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

Vervain Ex. 2011 p. 66 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 66
Micron v. Vervain
IPR2021-01550

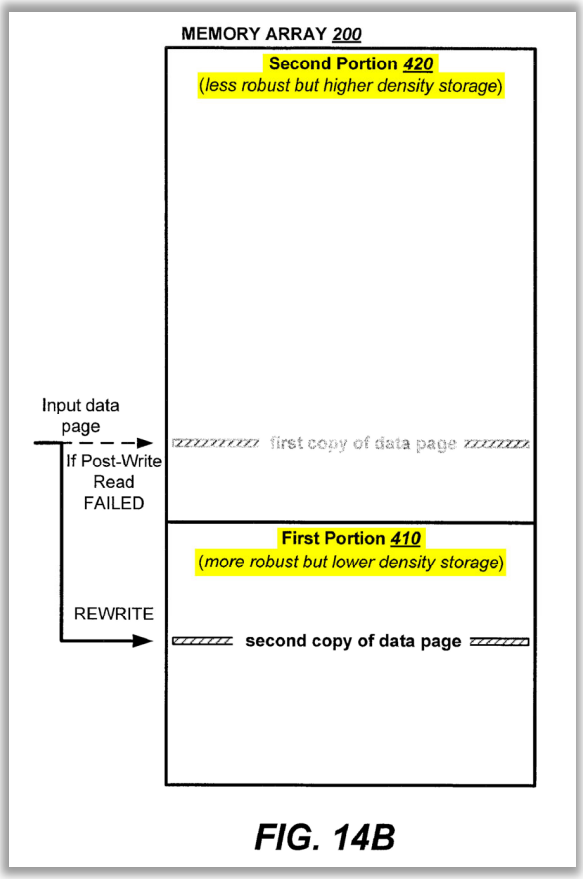


FIG. 14B

Dusija at Fig. 14B (annotated).

140. Dusija's "second portion" is a "*multilevel cell (MLC) memory space*." Specifically, Dusija explains that "[t]he second portion 420 has the memory cells configured as high-density storage with each cell storing multiple bits of data," *i.e.*, MLC. Dusija at [0109].

141. Dusija's "first portion" is a "*single level cell (SLC) memory space*." Specifically, Dusija explains that "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 at [0109].

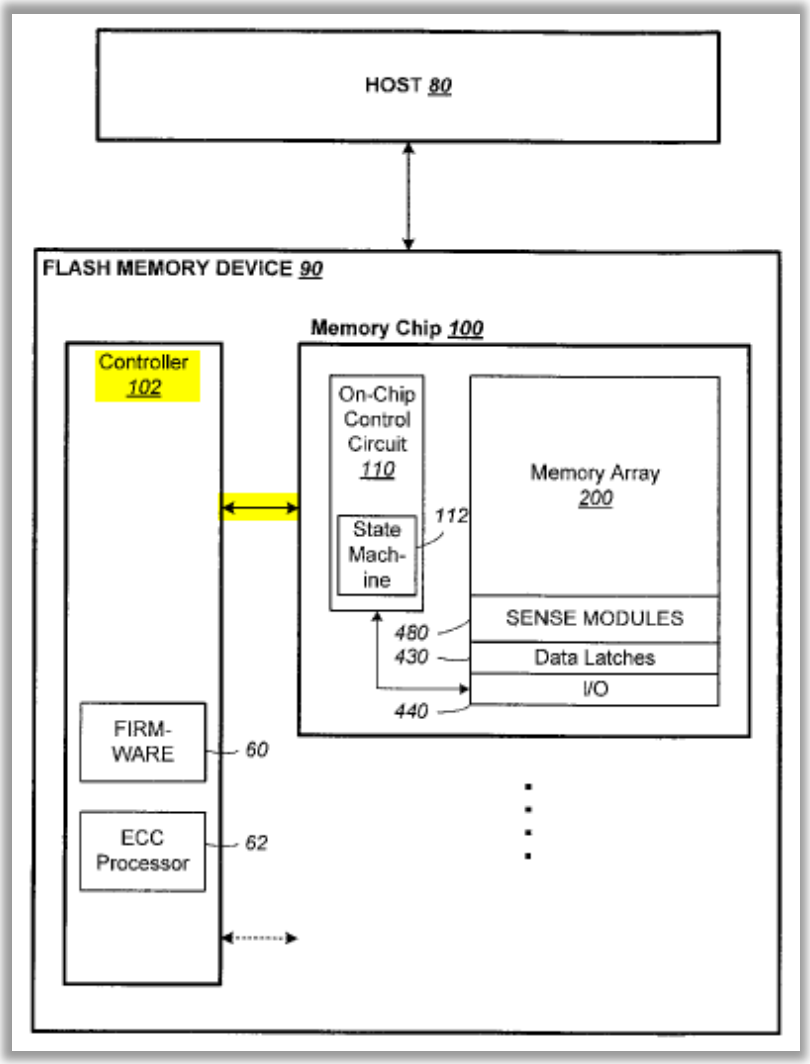
142. As I have explained above, in Section VII.A.5, a "memory space" is simply a collection of memory locations. Thus, Dusija's first portion—which is SLC is a "*single level cell (SLC) memory space*." Similarly, Dusija's second portion—which is MLC is a "*multilevel cell (MLC) memory space*."

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

143. Dusija discloses this limitation.

144. 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]):

Vervain Ex. 2011 p. 68 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 68
Micron v. Vervain
IPR2021-01550



Dusija at Fig. 1 (annotated), [0059], [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.* at [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]."

145. 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 at [0062] ("In another embodiment, the controller 102 is implemented within the host."). Dusija's statement "within the host" refers to, for example, performing controller operations using a computer's CPU and connecting to the flash memory system through, e.g., a bus.

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

146. Dusija discloses this limitation.

147. 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* Section IX.B.1.c (limitation [1.A.2]).

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

149. This first way is that Dusija's flash memory device uses a "block management system" that "maps and stores [sectors] to the physical structure of the memory array." Dusija at [0060]. Dusija accomplishes this through a "directory . . . embodied in the firmware of the controller" that maps logical addresses to physical addresses. *Id.* at [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

Vervain Ex. 2011 p. 70 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 70
Micron v. Vervain
IPR2021-01550

spaces (the "section portion 420" and "first portion 410") respectively. For example, Dusija discloses updating a 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). *Id.* at [0117], [0129], [0138]. A POSA would recognize that Dusija's disclosure that "a directory . . . will be updated to direct subsequent access to the second copy" means that Dusija's logical to physical address map will be updated to map a particular logical address (or addresses) to particular physical locations in Dusija's MLC memory elements. A POSA would recognize that, taken together, the physical locations in Dusija's MLC memory elements are "*the MLC memory space*." *See* Section VII.A.5 (noting the ordinary meaning of the phrase "memory space").

150. Paley, which is incorporated into Dusija (Dusija at [0061]), further explains that "mapping information is maintained in a set of logical to physical directories." Ex. 1028, Paley at [0162]. Paley illustrates such a directory to map logical to physical addresses:

Vervain Ex. 2011 p. 71 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 71
Micron v. Vervain
IPR2021-01550

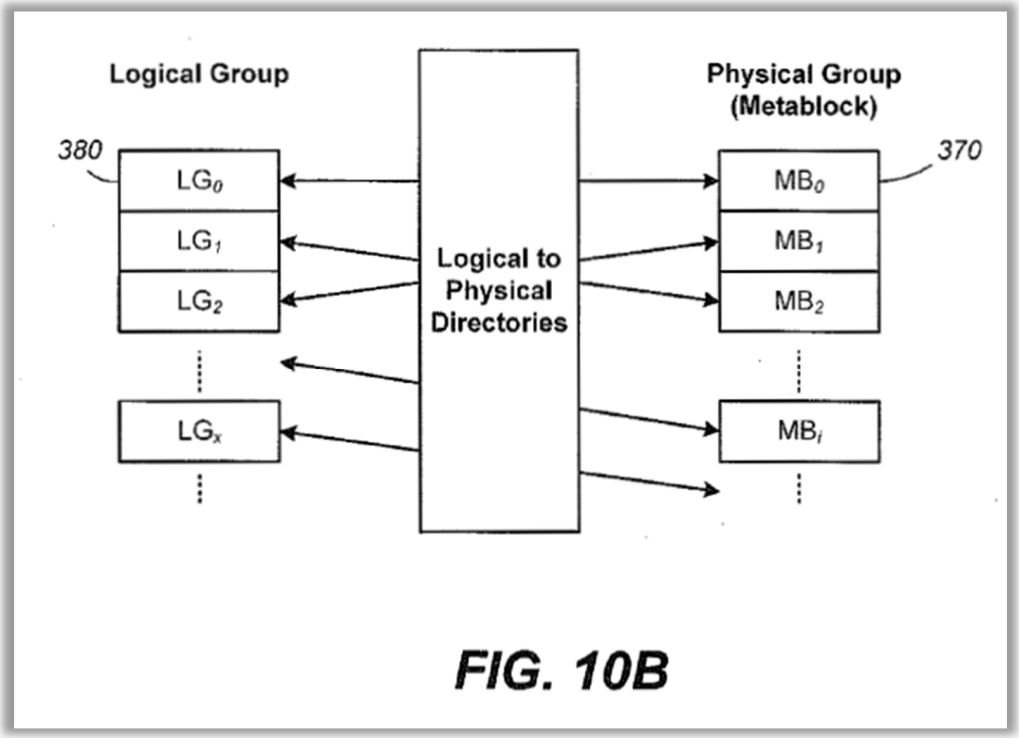


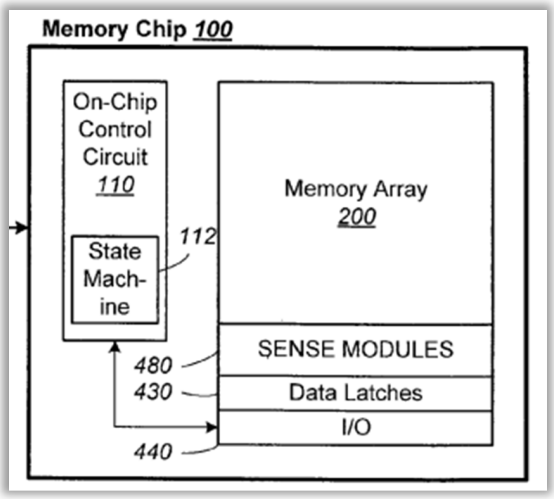
FIG. 10B

Id. at Fig. 10B.

151. Accordingly, Dusija discloses that the MLC memory elements "*can be mapped into the MLC memory space*."

152. The second way that Dusija discloses this limitation is 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 starts by noting that its memory "includes a memory array 200 of memory cells with each cell capable of being configured as a multi-level cell ('MLC') for storing multiple bits of data." Dusija at [0059]. It then discloses "[c]onfiguring the memory into first and second portions, the first portion having memory cells operating with a margin of error larger than that of

the second portion." *Id.* at [0120]. For example, the memory array 200 starts off with "memory cells with each cell capable of being configured as a multi-level cell ('MLC')":



Id. at [0059], Fig. 1 (excerpted).

153. Then, it "configures" the memory array 200 into a "first portion 410" (MLC) and "second portion 420" (SLC):

Vervain Ex. 2011 p. 73 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 73
Micron v. Vervain
IPR2021-01550

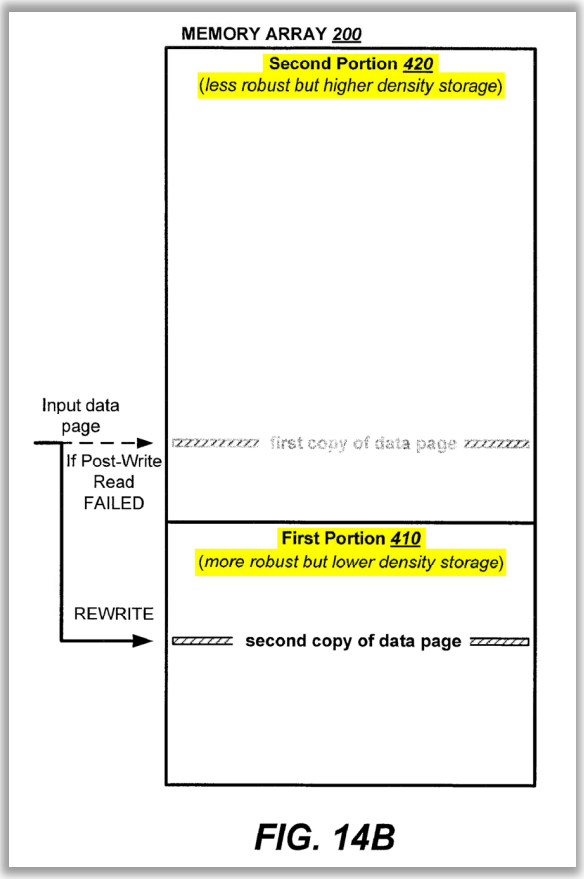


FIG. 14B

Id. at Fig. 14B (annotated), [0109], [0120].

154. Thus, in this second way, 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" respectively).

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

155. Dusija discloses this limitation.

156. 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].

157. Second, also as discussed above, Dusija discloses that the SLC memory elements "*can be mapped into the SLC memory space*" through (1) use of a directory that maintains a logical-to-physical address map for the SLC memory space ("first portion 410") (Dusija, [0117], [0129]) and (2) configuring the memory to be SLC and assigning them to the SLC space ("first portion 410") (Dusija, [0059], [0109], [0120]). Limitation [1.C]. A POSA would have recognized that just as Dusija discloses an MLC memory space, limitation [1.C], so too does Dusija disclose an SLC memory space.

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

158. 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).

Vervain Ex. 2011 p. 75 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 75
Micron v. Vervain
IPR2021-01550

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"

159. I explain what an FTL is in the technical background section above. *See* Section VII.A.6. Dusija discloses, or at least renders obvious, this limitation.

160. 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, *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.

161. First, as Sections IX.B.1.e-f (limitations [1.C-D]) explain, Dusija's flash memory device uses a "block management system" that "maps and stores [sectors] to the physical structure of the memory array." Dusija at [0060]. More specifically, Dusija performs this using a "directory . . . embodied in the firmware of the controller." Dusija at [0117]. A POSA would have immediately understood the "directory" to be a logical-to-physical address map. I discuss such addressing in Section VII.A.5. That is, Dusija's controller and directory that map logical to physical addresses are, together, "*an FTL flash translation layer*."

Vervain Ex. 2011 p. 76 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 76
Micron v. Vervain
IPR2021-01550

162. While Dusija does not expressly use the words "FTL," its disclosures would have been understood to include an FTL, and the use of an FTL in Dusija's system would certainly have been obvious. A POSA would have described Dusija's directory and controller to operate as an "FTL." Ex. 1019, Micheloni at 40, 492 ("when the file system sends the command to write or update a certain logical page, the FTL actually writes it in a different available physical page and updates a table storing the logical to physical mapping (mapping table)"). Moreover, as I explained in the background section, an FTL was conventionally included in flash memory devices in order to map logical addresses from a host file system to physical addresses of the flash memory. *See* Section VII.A.6.

163. Second, Dusija's "directory," *i.e.*, the logical-to-physical address map, is "*an address table*." Dusija at [0117], *see also id.* at [0060]. A POSA would have understood that a logical-to-physical address map is implemented as a table (*i.e.*, where one column corresponds to a logical address range, another column corresponds to a physical address range, and each row represents a particular page). Ex. 1028, Paley at Fig. 10B, [0162]; *see also* Ex. 1019, Micheloni at 492 (describing that the "logical to physical mapping" is stored in a "mapping table"). Indeed, Gorobets II,[7] which is incorporated by reference into Paley (Ex. 1028,

[7] US 8,250,333.

Vervain Ex. 2011 p. 77 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 77
Micron v. Vervain
IPR2021-01550

Paley at [0003]), explains that "[a]n address table in the memory system includes the mapping of the logical address space to the physical memory. In particular, the address table includes pages indicating the mapping of logical groups to physical blocks in the memory system." Ex. 1053, Gorobets II at 1:32-36.

164. Third, Dusija maintains its "directory" "in a block management system embodied in the firmware of the controller." Dusija at [0117]. That is, the controller (which itself implements the FTL) maintains the address table ("*wherein the at least one controller, or FTL, or a combination of both maintain an address table*"). *Id.* Note that Dusija also incorporates by reference Paley (Dusija at [0061]), which provides more details on an exemplary block management system. Ex. 1028, Paley at Fig. 10B, [0162], [0171] (pages that includes "SLC" and "MLC"), [0154]-[0155], [0159] (controller maintains map to map logical addresses, which the host uses, to physical addresses), [0020] (describing as background logical-to-physical mapping).

165. Fourth, Dusija discloses, or at least renders obvious, maintaining its "directory" ("*address table*") "*in one or more of the memory elements and random access volatile memory*." To start, as Section IX.B.1.b, limitation [1.A.1] illustrates, Dusija would have been understood (or it would have been obvious) to include RAM in its controller. In view of this, a POSA would have understood that Dusija stores the address map in this RAM because, for example, (1) Dusija

Vervain Ex. 2011 p. 78 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 78
Micron v. Vervain
IPR2021-01550

discloses maintaining its firmware (software that includes the "directory") in its controller (and thus would be in the controller's RAM during operation), and (2) Dusija describes its controller managing operations that involve the directory (and thus the controller would utilize the directory in its RAM). Dusija at [0059]-[0060], [0117]. Paley, which Dusija incorporates by reference (Dusija at [0061]), confirms this understanding, as it discloses that a logical-to-physical map (*i.e.*, Dusija's directory) "is set up in the controller RAM." Ex. 1028, Paley at [0021].

166. 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 well. As discussed above, Dusija fully incorporates the disclosures of Paley (Dusija at [0061]), which, in turn, incorporates WO 03/027828 to Gorobets et al. (Ex. 1050, Gorobets WO), stating that "WO 03/027828 [Gorobets WO] . . . 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." Ex. 1028, Paley at [0023]. In particular, Gorobets WO discloses storing mapping information in RAM with a "master table" of the mapping information in Flash, which protects against inadvertent power supply loss. Ex. 1050, Gorobets WO at 26:11-27:21, 2:12-19. Indeed, it was well-known that it was optimal to store address information in faster RAM, while backing up the contents in Flash in case of power loss. Ex. 1033, S. Lee at

Vervain Ex. 2011 p. 79 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 79
Micron v. Vervain
IPR2021-01550

[0169] (transferring book-keeping records from Flash to RAM); [0159] (book-keeping records include physical-to-logical address maps and backing up in Flash enables "Power-Failure Recovery"). Thus, at the very least, a POSA would have been motivated to implement Gorobets' method of storing a directory (mapping information) in nonvolatile memory elements based on Dusija, Paley, and/or the knowledge of a POSA, as such practice was routine as basic data back-up protection.

167. 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"

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

169. 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. Dusija at [0059], [0060], Fig. 1. Thus, a POSA would recognize that

Vervain Ex. 2011 p. 80 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 80
Micron v. Vervain
IPR2021-01550

Dusija's controller "*control[s] access of the MLC and SLC nonvolatile memory elements . . . for storage . . . therein*."

170. Second, as Section IX.B.1.b (limitation [1.A.1]) illustrates, Dusija would have been understood (or it would have been obvious) to use RAM in conjunction with its controller. It would have been understood, and certainly obvious, that Dusija's controller manages access to this RAM. To start, given that the RAM would be internal to the controller, it would have been understood that the controller would directly access it. Moreover, Dusija discloses maintaining its firmware (software that includes the "directory") in its controller, and Dusija discloses its controller managing operations that involve the directory. Dusija at [0059]-[0060], [0117]. In other words, the controller must directly access its RAM to store and carry out these functions with the directory. This is because, as discussed above, the directory is maintained in RAM. Further, Dusija incorporates by reference Paley (Dusija at [0061]), and Paley discloses RAM in its controller which the controller accesses. Ex. 1028, Paley at [0119]; *see also* Ex. 1049, Harari at 13:24-42 (discussing "controller" controlling access to "volatile cache memory"). It was well known that a controller controls access to its internal memory. Ex. 1019, Micheloni at 516. Accordingly, Dusija discloses, or at least renders obvious, "*the controller controlling access of … the random access volatile memory for storage of data therein*."

Vervain Ex. 2011 p. 81 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 81
Micron v. Vervain
IPR2021-01550

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;"

171. 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 at [0111]-[0116], [0119]-[0124].

172. 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" (from the host) to the "second portion," *i.e.*, MLC. Dusija at [0111]; Section IX.B.1.c, limitation [1.A.2] ("second portion" is MLC). The controller manages this operation, i.e., Dusija's controller receives the host write operations and manages the write operation to the physical memory. Dusija at [0060] (controller "controls" and "manages" memory operations).

173. 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 at [0112]-[0115]. Since it would have been obvious to implement Dusija's cache in volatile memory, *see* Section IX.B.1.b (limitation [1.A.1]), a

Vervain Ex. 2011 p. 82 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 82
Micron v. Vervain
IPR2021-01550

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*." In other words, this "cache" would have been understood to be Dusija's controller RAM that it also uses to, e.g., execute firmware and buffer incoming writes. Sections IX.B.1.b and IX.B.1.g, demonstrate that at a minimum, it would have been obvious to implement Dusija's "cache" as random access volatile memory.

174. Separately, Dusija discloses this same process as part of a "post-write read and adaptive rewrite." Dusija at [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;"

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

176. First, as I explained above for limitation [1.G.2], 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*"). Section IX.B.1.j; Dusija at [0111]-[0116], [0119]-[0124]. Note that Dusija

Vervain Ex. 2011 p. 83 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 83
Micron v. Vervain
IPR2021-01550

rewrites the data if it detects too many errors (if the comparison determines a failure of the data integrity test). *Id.* at [0116], [0124].

177. 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 at [0112] ("[l]ater, 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. In other words, after the write is complete (and usually as time permits as I describe with respect to claim 11), the post-write read occurs.

178. 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 doing so would have been obvious). Recall that data is retained in Dusija's "cache," which would have been understood, and at least obvious, to be RAM. *See* Section IX.B.1.b. As Paley, Harari, and Micheloni illustrate, a controller typically has a RAM as a cache and to support other processing. *Id.* Indeed, as I have explained above, it is well known for a controller to use such RAM for data processing that requires quick memory accesses, like the processing necessary for Dusija's data integrity test. *Id.*; Sections VII.A.4, IX.B.1.b, IX.B.1.g, limitations [1.A.1, E]. Again, it is the controller's RAM that the controller uses to perform such processing.

Vervain Ex. 2011 p. 84 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 84
Micron v. Vervain
IPR2021-01550

179. Dusija's "post-write read" functionality may be implemented after every write. For example, Dusija discloses that, if the Hot_count_enable_flag is not enabled, "EPWR is implemented from the beginning of life of the memory device." Dusija at [0203]. A POSA would have understood that Dusija's Hot_count_enable_flag resided in the device's file system configuration file and could have been modified by a user to ensure that the data integrity test would have been enabled at the beginning of the device's lifetime.

180. Thus, a POSA would recognize that Dusija discloses performing the data integrity test "*by comparing the stored data to the retained data in the random access volatile memory*." *See* Dusija at [0112] (describing that "the data page is read back in a 'post write read' to determine if there are any errors. This [can be] accomplished . . . by comparison with the original copy which may be cached."), [0122]-[0123].

181. Dusija's controller, using its RAM, would be understood to perform this operation because Dusija's controller manages memory operations. Dusija, Fig. 1, [0062].

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"

182. Dusija discloses this limitation.

Vervain Ex. 2011 p. 85 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 85
Micron v. Vervain
IPR2021-01550

183. 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* Sections IX.B.1.e, g, limitations [1.C, F]. Again, this directory is the logical-to-physical address map.

184. 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 the lifetime (endurance)":

> The controller conducts a data integrity test after every write. If a particular address range fails a data integrity test, the address range is remapped from MLC NAND flash to SLC NAND flash. As the SLC NAND flash is used to boost the lifetime (endurance) of the storage system, it can be considerably lesser in amount than the MLC NAND flash.

300 Patent at 5:14-20; *see also id.* at 1:45-52 (increasing lifetime by using MLC and SLC). A POSA would understand that one way of maximizing lifetime is to remap the data into SLC NAND flash. I discuss that SLC is more robust than MLC and that this extends the lifetime of the device by writing to more robust areas. Section VII.A.7.

Vervain Ex. 2011 p. 86 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 86
Micron v. Vervain
IPR2021-01550

185. Just like the 300 Patent, as Section IX.B.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*"). As it was known and as Dusija discloses, 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 at Abstract, [0109]. SLC provides a portion of memory that will last longer than MLC. Thus, writing to SLC when MLC fails, as Dusija discloses, will maximize lifetime because the SLC is more robust. *Id.*; *see also id.* at [0110] (recognizing that SLC cells provide for a "longer operational life"). In other words, first, the SLC is more likely to be unworn because it has a longer lifetime and second, the data will then reside in memory that has longer life.

186. Note that, even aside from Dusija's data integrity test feature, 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, thereby maximizing lifetime by using all possible blocks until complete failure. Dusija at [0162]; Ex. 1019, Micheloni at 42 (disclosing bad block management). 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

Vervain Ex. 2011 p. 87 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 87
Micron v. Vervain
IPR2021-01550

robust), but if the MLC blocks fail, maintaining the logical to physical mapping to SLC. Dusija at [0166]-[0174]. This too would maximize lifetime by avoiding less-reliable MLC blocks. Both of these techniques, however, ensure that even if a bad memory element arises, alternatives are available rather than simply having the operation fail.

187. Third, Dusija's "directory" is a logical-to-physical address table. *See* Sections IX.B.1.e, g (limitations [1.C, F]). Dusija discloses that the mapping may include "virtual [] blocks" (*i.e.*, "*blocks*"), Dusija at [0167], and "page[s]" in memory (*i.e.*, "*pages*"). *Id.* at [0012] ("multi-sector page"), [0111], [0117] (incoming logical pages written to physical MLC/SLC pages). Paley, which Dusija incorporates by reference (*id.* at [0061]), expressly contemplates block-level addressing. Ex. 1028, Paley at [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*."

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."

188. Dusija discloses this limitation.

Vervain Ex. 2011 p. 88 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 88
Micron v. Vervain
IPR2021-01550

189. First, Dusija discloses that data is first written to the "second portion". Dusija at [0111] (*i.e.*, MLC, *id.* at [0109]). Then, the data is read back from MLC for a comparison with an original cache copy to check for errors. *Id.* at [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". *Id.* at [0116] (*i.e.*, SLC, *id.* at [0109]). 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.* at [0117]. This updates the logical-to-physical address map, i.e., the logical address becomes associated with the new physical SLC address. 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]

[8] The 300 Patent uses "address ranges" to refer to basic flash addressing, *e.g.*, a block, page, or sector address. 300 Patent at Figs. 2a-b, 5:54-58 (explaining minimum quanta of address, *e.g.*, a block or page), 3:7-11 (sector). I have explained more regarding address ranges in the technology background section. *See* Section VII.A.5.

Vervain Ex. 2011 p. 89 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 89
Micron v. Vervain
IPR2021-01550

190. 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 at [0116]. 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 original MLC addresses that failed the test). A POSA would recognize, for example, that rewriting a block previously in a second portion (*i.e.*, MLC) to a first portion (*i.e.*, SLC) would necessarily involve remapping the block to a new physical address range. Again, this process involves updating the logical-to-physical map (remapping the logical address from being associated with a physical MLC address to a physical SLC address). Finally, as Section IX.B.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 allow the data now stored in SLC to achieve greater endurance. Section VII.A.7.

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

Vervain Ex. 2011 p. 90 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 90
Micron v. Vervain
IPR2021-01550

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."

192. Dusija discloses, or at least renders obvious, this limitation by the disclosure of its "directory," which is a flash translation layer. *See* Section IX.B.1.h, limitation [1.F]. Further, Dusija discloses that its "directory" is "embodied in the firmware of the controller." *E.g.*, Dusija at [0117]. Firmware, by definition, is software. Ex. 1054, Microsoft Computer Dictionary at 215 (defining "firmware" as "[s]oftware routines"). Dusija also incorporates by reference Paley, which discloses that ROM stores its firmware, *i.e.*, read-only executable software code. Ex. 1028, Paley at [0119]. Paley further explains that the "memory manager . . . is a software component." *Id.* at [0153]. Thus, Dusija's "directory" ("*FTL flash translation layer*") would be understood to be a "*software module*."

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."

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

194. As an initial matter, 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 at 6:15-32. Dusija discloses or renders obvious the same disclosure.

Vervain Ex. 2011 p. 91 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 91
Micron v. Vervain
IPR2021-01550

195. As Section IX.B.1.b (limitation [1.A.1]) demonstrates, Dusija discloses or renders obvious "*random access volatile memory*" being "*embedded*" in the "*controller*." As that section describes, Dusija discloses controller 102, which includes firmware 60 that manages memory operations. Dusija at Fig. 1, [0059], [0062]. A POSA would have understood that Dusija's controller includes RAM in order 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. Ex. 1028, Paley at Fig. 1, [0119]. Moreover, it would have been obvious to modify Dusija to embed RAM into the controller because doing so was conventional. Ex. 1019, Micheloni at 132, 144.

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

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

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

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

198. As Section IX.B.1.b (limitation [1.A.1]) demonstrates, Dusija discloses or renders obvious "*random access volatile memory*" for use as Dusija's

Vervain Ex. 2011 p. 92 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 92
Micron v. Vervain
IPR2021-01550

cache. As Section VII.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." 300 File History at 48, 76. 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 VII.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.

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

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

200. As Section VII.B.1.b (limitation [1.A.1]) demonstrates, Dusija discloses or renders obvious "*random access volatile memory*" for use as Dusija's cache. As Section VII.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 an SRAM cache as an option. Dusija also incorporates by reference Paley, which discloses using SRAM as a volatile memory. Ex. 1028, Paley at [0423]-[0425], [0502]-[0503]

Vervain Ex. 2011 p. 93 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 93
Micron v. Vervain
IPR2021-01550

201. Moreover, the Examiner repeatedly took official notice that SRAM and DRAM are "volatile memory types" that are often used as "cache memory." 300 File History at 48, 76. 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 VII.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.

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."

202. Dusija discloses this limitation.

203. As Section IX.B.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 at [0109]) and updating the address map in the controller ("updat[ing]" the "directory"). *Id.* at [0116]-[0117]. A POSA would understand that Dusija's controller performs these operations. *Id.* at [0060] (the controller "controls and manages higher level memory operations").

Vervain Ex. 2011 p. 94 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 94
Micron v. Vervain
IPR2021-01550

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

204. Dusija discloses this limitation.

205. Dusija discloses that memory cells operating as SLC have a "longer operational life," *i.e.*, "*higher endurance*," than MLC memory cells. Dusija at [0110] (recognizing that SLC cells provide for a "longer operational life"). It was well-known to a POSA that SLC cells have higher endurance than MLC cells. Ex. 1019, Micheloni at 511. In addition, the 300 Patent admits it was background knowledge that SLC cells have higher endurance than MLC cells. 300 Patent at 3:41-44.

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."

206. Dusija discloses this limitation, namely, "[t]he second portion 420 [MLC] has the memory cells configured as high density storage with each cell storing multiple bits of data." Dusija at [0109]. Dusija's MLC is multilevel cell memory. *Id.* at [0059].

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

207. Dusija discloses this limitation, namely, "[t]he second portion 420 [MLC] has the memory cells configured as high density storage with each cell

Vervain Ex. 2011 p. 95 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 95
Micron v. Vervain
IPR2021-01550

storing multiple bits of data." Dusija at [0109]. Dusija's MLC is multilevel cell memory. *Id*., [0059].

11. Claim 12:

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

208. To the extent the preamble is limiting, the preamble is identical to, and satisfied for the same reasons, as [1.PRE]. *See* Section IX.B.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;"

209. 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 IX.B.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;"

210. 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 IX.B.1.d, h.

Vervain Ex. 2011 p. 96 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 96
Micron v. Vervain
IPR2021-01550

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;"

211. These limitations are satisfied for the same reasons as, and are nearly identical to, limitations [1.C-D]. *See* Sections IX.B.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."

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

212. This limitation is satisfied for the same reasons as, and is identical to, limitation [1.E]. *See* Section IX.B.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;"

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

Vervain Ex. 2011 p. 97 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 97
Micron v. Vervain
IPR2021-01550

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;"

214. This limitation is satisfied for the same reasons as, and is identical to, limitation [1.H]. *See* Section IX.B.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"

215. This limitation is satisfied for the same reasons as, and is identical to, limitation [1.I]. *See* Section IX.B.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."

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

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

217. 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

Vervain Ex. 2011 p. 98 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 98
Micron v. Vervain
IPR2021-01550

alone, it is obvious in view of Dusija in combination with Sutardja. It would have been obvious to adapt Dusija in view of Sutardja's teachings to achieve the claimed inventions. In fact, as I will discuss in Section IX.C.2, Dusija already has the building blocks to implement these features of 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."

218. Dusija in view of Sutardja renders this limitation obvious.

219. As an initial matter, note that the 300 Patent uses "accessed" to at least include the number of times to which the blocks are written. 300 Patent at 6:59-62 ("maintain a count of those blocks that are accessed (written to) most frequently").

220. Sutardja's "data shift analysis" discloses this limitation. Sutardja at [0148]-[0149]; *see also id.* at [0128]. When the time to perform the 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" ("*frequently accessed portions of the memory space*"). *Id.* at [0149], [0146] (logical addresses having high write frequencies (*i.e.*, "frequently accessed") are those with write counts greater than the threshold); *see also id.* at Fig. 7C, [0106] (first memory is MLC; second memory is SLC). If the number of writes is greater than the threshold, the "control

Vervain Ex. 2011 p. 99 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 99
Micron v. Vervain
IPR2021-01550

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." *Id.* at [0149]. By Sutardja disclosing remapping the logical address of MLC to SLC, it would have been understood that it must copy the data to SLC as well, which Sutardja describes in paragraph 167 ("swapping data"). *Id.* Note that Sutardja performs this process "*periodically*." *Id.* at [0148] ("In FIG. 7B, control determines whether time to perform data shift analysis has arrived in step 514."). In fact, Sutardja expressly discloses that its wear-leveling techniques (which its "data shift" is, *id.* at [0148]) are applicable "periodically." *Id.* at [0167]. Accordingly, Sutardja discloses "*wherein the contents of frequently accessed portions of the memory space* [frequently written MLC blocks] *are periodically moved from the MLC space to the SLC space* [remapping and copying the data from MLC to SLC]."

221. Note that Sutardja's "data shift" involves transferring the contents of MLC blocks to SLC blocks without the data changing, *i.e.*, not in connection with a host write operation. Sutardja at [0148]-[0149]. In other words, Sutardja discloses that the "data shift" transfers the data independent of any host write operation, *i.e.*, it merely copies the data from MLC memory to SLC memory. *Id.* Through this operation, the data shift simply transfers the data in the MLC block to the SLC block and updates the mapping table without the data changing. *Id.*; *see also id.* at [0167] (describing "swapping data").

Vervain Ex. 2011 p. 100 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 100
Micron v. Vervain
IPR2021-01550

222. In addition, and unlike the "data shift," Sutardja discloses a second way to use the frequency count to transfer 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 above a threshold, the controller may direct the write to a physical SLC block, thereby transferring the data from MLC memory space to SLC memory space. Sutardja at [0146]-[0147]. In this case, before the write, a logical address is associated with a physical address that stores the data. After the write, the logical address is remapped to the new physical address. In this way, the data corresponding to the logical addresses is transferred from the MLC to SLC block (note, the data will be different assuming the written data is different than the original data).

223. As I explained above, *see* Section VIII.C, the ordinary meaning of the term "periodically" includes events that occur from time to time. Sutardja discloses that its system "determines whether [the] time to perform data shift analysis has arrived in step 514." Sutardja at [0148]. In other words, the move occurs periodically, *e.g.*, when the device is not otherwise busy. *Id.* 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." *Id.* at [0167] (emphasis added). The data analysis is a wear-leveling technique. *Id.* at [0148].

Vervain Ex. 2011 p. 101 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 101
Micron v. Vervain
IPR2021-01550

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 on a periodic basis.

224. It was further well known to a POSA (and obvious) that when the time has arrived to perform the data transfer it may be measured off an overall access (*e.g.*, write count). For example, U.S. Patent Application Publication No. 2010/0017650 to Chin et al. discloses such a period. Chin discloses its controller performing wear leveling "depending on an erase count associated with each region" (*i.e.*, a collective count for the MLC region). Ex. 1030, Chin at [0018], [0074] (segment erase count can be total erase count of all blocks in that segment). Similarly, European Patent 2,291,746 to Radke et al. discloses a flash memory system, complete with MLC and SLC, in which its controller moves data from MLC to SLC once the usage data exceeds a threshold value. Ex. 1031, Radke at [0022], [0026].

225. Thus, Sutardja's disclosure of directing high write blocks to SLC satisfies the limitation that "*the contents of frequently accessed portions of the memory space*" [high write counts blocks] "*are periodically*" [upon a write operation that reaches the threshold] "*moved from the MLC space to the SLC space*" [the logical address is remapped from MLC space to SLC space].

Vervain Ex. 2011 p. 102 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 102
Micron v. Vervain
IPR2021-01550

2. Motivation to Combine

226. 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.

227. First, both Dusija and Sutardja describe techniques for improving the performance and reliability of flash memory systems that include both MLC and SLC memory. Dusija at [0018]; Sutardja at Abstract, [0012]. It was well known in the art that a flash memory controller could implement multiple techniques that each improve the performance or reliability of the flash memory device. For example, the same controller might implement techniques for wear leveling (both dynamic and static), garbage collection, bad block management, and error correction. Ex. 1019, Micheloni at 39. 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. And with Dusija already having both MLC and SLC, a POSA would have been motivated to consider other techniques applicable to MLC and SLC memories and implement those techniques that would further improve the system.

228. Second, Sutardja's rationale for periodically remapping hot blocks to SLC—to optimize cost and memory lifetime (Sutardja at [0102])—applies equally to Dusija's system, which also seeks to maximize memory lifetime. That is, by

Vervain Ex. 2011 p. 103 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 103
Micron v. Vervain
IPR2021-01550

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 at [0018]; Sections VII.A.2, 6, 7. In other words, a POSA would have recognized that Sutardja and Dusija each disclose different and complementary improvements to the reliability of flash memory devices. And since Dusija already has both MLC and SLC, it would be natural to further Dusija's objectives with Sutardja's hot-blocks function.

229. 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. Ex. 1019, Micheloni at 511. Thus, a POSA would have recognized that integrating Sutardja's teachings into Dusija would further improve the speed of Dusija's system.

230. 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. In other words, given that a small fraction (e.g., 20%) of the data is typically frequently written, it was understood that this data should be kept (and allocated) to the SLC, while the other data is kept in the less expensive and higher density MLC. That is to say, this use

Vervain Ex. 2011 p. 104 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 104
Micron v. Vervain
IPR2021-01550

of SLC and MLC is a common sense application given the normal use of flash memory systems. And a POSA would have recognized that this would maximize performance and cost.

231. Fifth, a POSA would have had a high expectation of success in adapting Dusija to include Sutardja's hot blocks teaching. Dusija already discloses maintaining an erase count for blocks, illustrating that such counting was routine and applicable to Dusija. Dusija at [0154]. Thus, counting blocks to determine the most frequently accessed blocks was, in effect, already done by Dusija. Dusija also explicitly recognizes that other features and modifications are implementable within its system, suggesting to a POSA that other improvements are and should be made. *Id.* at [0151].

232. Further, Dusija already discloses memory management features that occur in the background, *e.g.*, counting erase cycles and enabling error management. Dusija at [0155]-[0160]. Thus, Dusija's system is compatible with implementing Sutardja's hot block allocating feature in its background processing. That is, when the time has arrived in Dusija as part of its background operations (as Sutardja teaches its data shift applies to), a POSA would expect that Sutardja's data shift is applicable to Dusija. Finally, as Section IX.B.1.m [1.J] explains, Dusija discloses rewriting data and remapping the logical address to a new physical address as part of a data integrity test. Sutardja similarly discloses

Vervain Ex. 2011 p. 105 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 105
Micron v. Vervain
IPR2021-01550

rewriting data and remapping addresses, but in the context of allocating frequently written blocks. Thus, Dusija already includes the building blocks (controller functionality to use a logical-to-physical map) for rewriting and remapping data to carry out Sutardja's allocation of frequently written blocks. A POSA would have a high expectation of success in implementing Sutardja's algorithms in Dusija's controller.

X. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

233. Having reviewed the file history and the other exhibits cited in this declaration, I am not aware of any objective indicia of non-obviousness for the 300 Patent's claims.

Vervain Ex. 2011 p. 106 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 106
Micron v. Vervain
IPR2021-01550

XI. DECLARATION

234. In signing this declaration, I understand that this declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the USPTO.

235. I am aware that willfully false statements and the like made by me in connection with this declaration are punishable by fine or imprisonment, or both (18 U.S.C. § 1001) and may jeopardize the validity of the Petition. All statements herein made by me of my own knowledge are true and all statements made on information and belief are believed to be true.

I declare under penalty of perjury that the foregoing is true and correct.

Executed this 21st day of September, 2021 at Fremont, CA.

Respectfully Submitted,

Dr. David Liu

Vervain Ex. 2011 p. 107 of 107
Vervain v. Phison
PGR2024-00048
Micron Ex. 1009, p. 107
Micron v. Vervain
IPR2021-01550