

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC. AND SAMSUNG ELECTRONICS CO., LTD.,
Petitioners,

v.

GAEA, LLC,
Patent Owner.

Case No. IPR2026-00425
U.S. Patent No. 10,776,023

**DECLARATION OF HEINER LITZ, PH.D.
IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 10,776,023**

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
II.	BACKGROUND	2
	A. Experience, Qualifications, and Status as an Independent Expert.....	2
	B. Applicable Legal Standards	4
	1. Anticipation	5
	2. Obviousness.....	5
III.	MATERIALS REVIEWED AND CONSIDERED	7
IV.	OVERVIEW OF THE TECHNOLOGY	11
	A. Continued Scaling of Storage Capacity and Performance.....	11
	B. Scaling Introduced New Device-Level Constraints.....	13
	C. Traditional Interfaces Hid Device Complexity.....	14
	D. Hyperscale Systems Drove Software-Programmable Policies	15
	E. Host-Programmable SSD Policies Were Known Before November 2016.....	16
	F. More-Complex Policies Used Request Attributes and Data- Processing Rules	18
	G. Policy-Programmable Storage at Multiple Layers.....	19
V.	THE '023 PATENT.....	21
	A. Overview	21
	B. Prosecution History.....	25

C.	Priority Date.....	27
D.	Person of Ordinary Skill in the Art.....	27
VI.	CLAIM CONSTRUCTION	28
VII.	THE PRIOR ART	29
A.	Olster (Ex. 1005).....	29
B.	Davis (Ex. 1008).....	32
C.	Hashimoto (Ex. 1006).....	34
D.	Kilian (Ex. 1007)	35
VIII.	OPINIONS ON GROUND 1: CLAIMS 1, 7, 9, 14, AND 15 ARE OBVIOUS IN VIEW OF OLSTER.....	35
A.	Independent Claims 1, 9:	39
1.	Preambles 1[pre]/9[pre]: “An apparatus comprising:”	39
2.	Limitations 1[a]/9[a]: “a storage media;”	41
3.	Limitations 1[b]/9[b]: “a device controller operatively coupled with the storage media configured to at least:”	43
4.	Limitations 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media;”	44
5.	Limitation 1[b.ii]: “receive a storage request, from the host, to store content;”	48
6.	Limitations 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”	49

7.	Limitations 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”	51
8.	Limitations 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content;”	54
9.	Limitations 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content;”	56
10.	Limitations 1[b.vii]/9[b.vi]: “recover the content from the storage media, [at least in part according to the storage information]/[according to the storage information and according to the at least one storage device policy]”	57
11.	Limitations 1[b.viii]/9[b.vii]: “respond to the content request at least by returning the content to the host.”	59
B.	Dependent Claims 7, 14, 15:	60
1.	Claim 7: “The apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”	60
2.	Claim 14: “The apparatus of claim 9, wherein the storage media comprises a solid-state memory device.”	62
3.	Claim 15: “The apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”	62
IX.	OPINIONS ON GROUND 2: CLAIMS 1-2, 7, 9, 14, AND 15 ARE OBVIOUS IN VIEW OF OLSTER IN COMBINATION WITH DAVIS	63
A.	Motivation to Combine	64

B.	Reasonable Expectation of Success	66
C.	Claims 1, 7, 9, and 14-15	69
D.	Dependent Claim 2	72
1.	Claim 2: “The apparatus of claim 1, wherein the at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types and to assign a customized recording method to each of the one or more content types.”	72
X.	GROUND 3: CLAIMS 1, 9, AND 14 ARE OBVIOUS IN VIEW OF HASHIMOTO	74
A.	Independent Claims 1, 9	78
1.	Preambles 1[pre]/9[pre]: “An apparatus comprising”	78
2.	Limitations 1[a]/9[a]: “a storage media;”	79
3.	Limitations 1[b]/9[b]: “a device controller operatively coupled with the storage media configured to at least:”	80
4.	Limitations 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media;”	84
5.	Limitation 1[b.ii]: “receive a storage request, from the host, to store content;”	91
6.	Limitations 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”	95
7.	Limitations 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”	97

8.	Limitations 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content;”	101
9.	Limitations 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content;”	103
10.	Limitations 1[b.viii]/9[b.vii]: “recover the content from the storage media, [at least in part according to the storage information]/[according to the storage information and according to the at least one storage device policy]; and”	104
11.	Limitations 1[b.ix]/9[b.viii]: “respond to the content request at least by returning the content to the host.”	107
B.	Dependent Claim 14:	108
1.	Claim 14: “The apparatus of claim 9, wherein the storage media comprises a solid-state memory device.”	108
XI.	GROUND 4: CLAIMS 7 AND 15-20 ARE OBVIOUS IN VIEW OF HASHIMOTO AND KILIAN	109
A.	Motivation to Combine	109
B.	Reasonable Expectation of Success	112
C.	Dependent Claims 7, 15	114
1.	Claim 7: “The apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”	114
2.	Claim 15: “The apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”	116
D.	Independent Claim 16	116

Declaration of Heiner Litz, Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 10,776,023

1.	Preamble 16[pre]: “An apparatus comprising:”	116
2.	Limitation 16[a]: “a storage media;”	116
3.	Limitation 16[b]: “a device controller operatively coupled with the storage media configured to at least:”	117
4.	Limitation 16[b.i]: “receive, from a host, a storage policy, wherein the storage policy comprises at least a write-protection policy and recording method for writing content to the storage media;”	117
5.	Limitation 16[b.ii]: “receive, from the host, a storage request to store content;”	119
6.	Limitation 16[b.iii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”	119
7.	Limitation 16[b.iv]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”	120
8.	Limitation 16[b.v]: “subsequent to storing the content, receive a delete request for the content, comprising at least the content identifier for the content;”	122
9.	Limitation 16[b.vi]: “in response to the delete request, retrieve from the memory of the device controller the storage information for the content; and”	123
10.	Limitation 16[b.vii]: “respond to the delete request, indicating that the content was not deleted in accordance with the write-protection policy.”	125
E.	Dependent Claims 17-20	127

1. Claim 17: “The apparatus of claim 16, wherein the storage policy and the delete request are received from the same source.” 127
2. Claim 18: “The apparatus of claim 16, wherein the storage policy and the delete request are received from different sources.” 128
3. Claim 19: “The apparatus of claim 16, wherein the delete request is a modification request.” 129
4. Claim 20: “The apparatus of claim 16, wherein the delete request is an overwrite request.” 130

XII. OPINIONS ON GROUND 2, CONTINUED: CLAIMS 32, 34, AND 36 ARE OBVIOUS IN VIEW OF OLSTER IN COMBINATION WITH DAVIS.....131

- A. Claim 32 131
 1. Preamble 32[pre]: “An apparatus comprising:” 131
 2. Limitation 32[a]: “a storage media;” 131
 3. Limitation 32[b]: “a device controller operatively coupled with the storage media configured to at least:” 131
 4. Limitation 32[b.i]: “receive, from a host, a storage device policy, wherein the storage device policy indicates recording methods for writing content to the storage media wherein each recording method corresponds to a content type;” 132
 5. Limitation 32[b.ii]: “identify the content type of a first content and the content type of a second content;” 133
 6. Limitation 32[b.iii]: “store the first content on the storage media by utilizing a first recording method of the

	recording methods in accordance with the storage device policy; and”	133
7.	Limitation 32[b.iv]: “store the second content on the storage media by utilizing a second recording method of the recording methods in accordance with the storage device policy wherein the first density recording method and the second recording method differ.”	134
B.	Dependent Claims 34, 36	136
8.	Limitation Claim 34: “The apparatus of claim 32, wherein the first content and the second content are both stored on the storage media.”	136
9.	Limitation Claim 36: “The apparatus of claim 34, wherein the first content and the second content are different content types.”	136
XII.	CONCLUSION	137

I. INTRODUCTION

1. I have been retained by Meta Platforms, Inc. (“Meta”) to analyze the scope and content of U.S. Patent No. 10,776,023 (“the ’023 patent”) relative to the state of the art at the time of the earliest application to which the ’023 patent claims priority. My analysis relates to claims 1-2, 7, 9, 14, 15-20, 32, 24, 26. I have also been retained to provide analysis regarding what a person of ordinary skill in the art (“POSITA”) would have understood at the time of the earliest application underlying the ’023 patent.

2. This declaration summarizes the opinions I have formed to date. I reserve the right to modify or amend my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this declaration, including in response to positions taken by Gaea, LLC (“Gaea”) or its experts that I have not yet seen, including any secondary considerations evidence that Gaea or its expert may consider and present.

3. It is my opinion that claims 1-2, 7, 9, 14, 15-20, 32, 34, and 36 of the ’023 patent are invalid as obvious based on the following grounds.

Ground 1	Claims 1, 7, 9, 14, and 15 are obvious in view of Olster
Ground 2	Claims 1, 2, 7, 9, 14, 15, 32, 34, and 36 are obvious in view of Olster and Davis
Ground 3	Claims 1, 9, and 14 are obvious in view of Hashimoto

Ground 4	Claims 7, 15-20 are obvious in view of Hashimoto and Kilian
----------	---

II. BACKGROUND

A. Experience, Qualifications, and Status as an Independent Expert

4. My curriculum vitae (“CV”) is Exhibit 1004.
5. A detailed list of my professional activities, memberships, and speaking engagements is included in my CV (Ex. 1004).
6. I am being compensated at the rate of \$600 per hour for my work on this declaration. My fee is not contingent on the outcome of any matter or on any of the technical positions I explain in this declaration.
7. I have no financial interest in Meta. I have been informed that Gaea, LLC (“Gaea”) claims ownership of the ’023 patent. I have no financial interest in Gaea.
8. I earned a degree in Computer Engineering from the University of Mannheim, Germany, in 2006. In 2012, I completed a Ph.D. in Computer Engineering from the same institution, graduating Summa Cum Laude. Between 2006 and 2012, my graduate research focused on computer architecture, hardware chip design, scalable storage systems, and high-performance computing networks.
9. In 2013, I began my academic career as a Postdoctoral Researcher at Stanford University, working in the research groups of Professors David Cheriton

and Christos Kozyrakis. From 2012 to 2016, my research centered on programmable “Open-Channel” solid-state drives (SSDs). I introduced techniques to reduce access latency by shifting performance-critical tasks such as data placement from the SSD to the host processor. This work led to “RAIL: Predictable, Low Tail Latency for NVMe Flash”, which achieved a 7× reduction in read access latency through host-orchestrated SSD data placement.

10. In 2017, I served as Visiting Faculty at Google, working on programmable storage devices within the Platforms Infrastructure Group.

11. In 2018, I joined the Computer Science Department at the University of California, Santa Cruz (UCSC), as an Assistant Professor. I was promoted to Associate Professor with tenure in 2023. In 2024, I was named the Kumar Malavalli Endowed Professor for Storage Systems Research, a position supported by a \$1 million gift from Kumar Malavalli.

12. Since 2023, I have served as Director of the Center for Research in Storage Systems (CRSS) at UCSC, a center comprised of five faculty and 15 Ph.D. students. CRSS advances research in storage systems and is supported by the National Science Foundation (NSF) and industry sponsors. Our work spans a range of topics, including programmable storage and emerging technologies such as Open-Channel and Zoned Namespace (ZNS) SSDs. In the context of ZNS, my lab

proposed the group append feature to optimize ZNS SSD integration with file systems (*e.g.*, “Append is Near: Log-based Data Management on ZNS SSDs”).

13. In 2024, I was a Visiting Professor at the Massachusetts Institute of Technology (MIT) in Cambridge.

14. Over the course of my academic career, I have published more than 80 peer-reviewed research papers, journal articles, and patents in the areas of storage systems and computer hardware. I have advised over 15 Ph.D. students and secured more than \$10 million in research funding. My honors include the NSF CAREER Award, Intel Outstanding Researcher Award, and multiple Faculty Awards from Google and Facebook. I received the Best Paper Award at MICRO 2023 and SoCC 2025 and several IEEE MICRO Top Pick awards. I have also served as Program Chair for HotChips 2023 and NVMW 2024, and as Associate Editor for IEEE Computer Architecture Letters (CAL).

B. Applicable Legal Standards

15. I understand that statutory and judicially created standards must be considered to determine the validity of a patent claim. I am not an attorney and, consequently, will offer no opinion on the law itself. My understanding of the pertinent law is described in this section and is the result of explanations provided by counsel. I have applied this understanding in my analysis.

16. I understand that a patent claim is unpatentable if it is anticipated by, or obvious in view of, the prior art. I further understand that the time frame of reference for determining whether a claim is unpatentable is from the perspective of a person of ordinary skill in the art at the time of invention.

1. Anticipation

17. I have been informed that for a patent claim to be anticipated by the prior art, each and every limitation of the claim must be found, expressly or inherently, in a single prior art reference as recited in the claim. I have been informed that a claim limitation not expressly found in a prior art reference is inherent if the prior art necessarily functions in accordance with, or includes, the claim limitation. Mere probability that a limitation is included is not sufficient to establish inherency.

2. Obviousness

18. In analyzing obviousness in light of the prior art I have been informed that it is important to understand the scope of the claims, the level of skill in the relevant art, the scope and content of the prior art, the differences between the prior art and the claims, and any objective indicia of non-obviousness (also called secondary considerations).

19. I have been informed that a patent claim is unpatentable for obviousness if the differences between the subject matter sought to be patented and the prior art

are such that the subject matter as a whole would have been obvious at the time the invention was made to a person of ordinary skill in the art to which said subject matter pertains. I have been informed that obviousness may be based on one reference or a combination of references. 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.

20. I have been informed that when a patented invention is a combination of known elements, the Patent Trial and Appeal Board (“Board”) must determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue by considering the teachings of prior art references, the effects of demands known to people working in a field or present in the marketplace, and the background knowledge possessed by a person of ordinary skill in the art.

21. I have been informed that the Supreme Court has recognized several rationales for combining references or modifying a reference to show obviousness of the claimed subject matter. I understand that several of these rationales are: (1) combining prior art elements according to known methods to yield predictable results; (2) simple substitution of one known element for another to obtain predictable results; (3) use of a known technique to improve a similar device (method or product) in the same way; (4) applying a known technique to a known device

(method or product) ready for improvement to yield predictable results; (5) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; (6) and some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

22. I have been informed that in order to prove that a claimed invention is not patentable for obviousness, a petitioner must (1) identify the differences between the claim and particular disclosures in the prior art references, singly or in combination, (2) specifically explain how the prior art references could have been combined in order to arrive at the subject matter of the claimed invention, and (3) specifically explain why a person of ordinary skill in the art would have had reasons to so combine the prior art references.

III. MATERIALS REVIEWED AND CONSIDERED

23. My opinions in this declaration are based on working in the field of data storage systems, including my teaching, research, and consulting. I have an established understanding of the relevant field at the relevant time frame and I have an understanding of the skill set, capabilities, and knowledge of a person of ordinary skill in the art as of 2016.

24. My opinions are also based on investigation and study of the patent at issue, its file history, and the prior art. In the course of forming my opinions, I have

Declaration of Heiner Litz, Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 10,776,023

reviewed the exhibits submitted with the Petition and any other materials cited in this declaration; I have reviewed the following exhibits, which I understand are attached to the Petition for *Inter Partes* Review of the '023 patent.

Exhibit No.	Description
1001	U.S. Patent No. 10,776,023
1002	File History of U.S. Patent '023 Patent
1005	U.S. Patent Application Publication No. 2015/0052167 to Olster et al. ("Olster")
1006	U.S. Patent Publication No. 2016/0313943 to Hashimoto et al. ("Hashimoto")
1007	U.S. Patent No. 9,075,851 to Kilian et al. ("Kilian")
1008	U.S. Patent No. 8,874,835 to Davis et al. ("Davis")
1009	U.S. Patent Application Publication No. 2012/0166749 to Eleftheriou et al. ("Eleftheriou")
1010	U.S. Patent Application Publication No. 2012/0246435 to Meir et al. ("Meir")
1011	U.S. Patent Application Publication No. 2015/0032938 to Salessi ("Salessi")
1012	U.S. Patent Application Publication No. 2015/0146364 to Chou ("Chou")
1015	U.S. Patent Application Publication No. 2014/0089278 to Lovinger et al. ("Lovinger")
1017	Chip Walter, 'Kryder's Law' (2005)' Scientific American, Aug. 2005.

Declaration of Heiner Litz, Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 10,776,023

Exhibit No.	Description
1018	Timothy Feldman and Garth A. Gibson, 'Shingled Magnetic Recording' ACM Queue / CMU, 2013.
1019	Albert Chen and Jim Malina, 'Addressing Shingled Magnetic Recording Drives with Linear Tape File System' SNIA Storage Developer Conference, 2013
1020	Jorge Campello, 'SMR: The Next Generation of Storage Technology' SNIA Storage Developer Conference, 2015.
1021	Samarth Bhargava, 'Lowering HAMR Near-Field Transducer Temperature via Inverse Electromagnetic Design' IEEE Transactions on Magnetics, Apr. 2015.
1022	Yu Cai et al., 'Read Disturb Errors in MLC Nand Flash Memory: Characterization, Mitigation, and Recovery' DSN 2015.
1023	Nitin Agrawal et al., 'Design Tradeoffs for SSD Performance' USENIX Annual Technical Conference, 2008
1024	Feng Chen, David A. Koufaty, and Xiaodong Zhang; 'Understanding Intrinsic Characteristics and System Implications of Flash Memory Based Solid State Drive' SIGMETRICS 2009
1025	Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, 'The Google File System' SOSP 2003
1026	Kirk McKusick and Sean Quinlan, 'GFS: Evolution on Fast-Forward,' ACM Queue, Aug. 2009 / Communications of the ACM, Mar. 2010
1027	Konstantin Shvachko et al., 'The Hadoop Distributed File System,' IEEE MSST 2010
1028	Sage A. Weil et al., 'Ceph: A Scalable, High-Performance Distributed File System' OSDI 2006
1029	Sage A. Weil et al., 'CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data' SC 2006

Declaration of Heiner Litz, Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 10,776,023

Exhibit No.	Description
1030	Leslie Ashton et al., ‘Two Decades of Policy-Based Storage Management for the IBM Mainframe Computer’ IBM Systems Journal, 2003
1031	IBM, ‘DFSMSHsm Primer’; IBM Redbooks, Sept. 2015
1032	IBM, ‘GPFS: Advanced Administration Guide’ including policy-based data management for GPFS, pre-2016 IBM documentation
1033	IBM, ‘IBM Spectrum Scale (formerly GPFS)’ IBM Redbooks, May 2015
1034	NVM Express, ‘NVM Express Revision 1.0’ Mar. 1, 2011
1035	NVM Express, ‘NVM Express Revision 1.2’ Nov. 3, 2014
1036	Open-Channel Solid State Drives, ‘NVMe Specification Revision 1.2’ Apr. 2016
1037	Trusted Computing Group, ‘Storage Security Subsystem Class: Opal Specification Version 2.00’ Feb. 24, 2012
1038	T10, “SCSI Block Commands - 3 (SBC-3),” Working Draft Revision 25, Oct. 27, 2010
1039	Fred Knight and David L. Black, ‘SBC-3 Thin Provisioning Commands’ T10 proposal 08-356r5, Jan. 14, 2009
1040	T10, ‘SCSI Block Commands - 4 (SBC-4)’ draft index, including pre-priority revisions in 2014-2016
1041	David A. Patterson, Garth Gibson, and Randy H. Katz, ‘A Case for Redundant Arrays of Inexpensive Disks (RAID)’ SIGMOD 1988
1042	Peter M. Chen et al., ‘RAID: High-Performance, Reliable Secondary Storage’ ACM Computing Surveys, 1994

Exhibit No.	Description
1043	John Colgrove et al., 'Purity: Building Fast, Highly-Available Enterprise Flash from Commodity Components,' SIGMOD 2015
1044	EMC Corporation, 'XtremIO Data Protection (XDP): Flash-Specific Data Protection,' White Paper, July 2014
1045	U.S. Patent Application Publication No. 2004/0230795 to Armitano ("Armitano")
1046	U.S. Patent Application Publication Nos. 2007/0244939 and 2007/0244940 to Devarakonda ("Devarakonda")
1047	U.S. Patent Application Publication No. 2011/0022642 to deMilo ("deMilo")
1048	U.S. Patent Application Publication No. 2010/0145917 to Bone ("Bone")
1049	U.S. Patent Application Publication No. 2010/0332401 to Prahlad ("Prahlad")
1050	U.S. Patent Application Publication No. 2014/0297695 to Dejana ("Dejana")
1051	IBM, 'Understanding Hard Drive Media Defects White Paper – Servers' IBM Product Documentation, Dec. 1997

IV. OVERVIEW OF THE TECHNOLOGY

A. Continued Scaling of Storage Capacity and Performance

25. Over the last several decades, storage capacity and performance have improved rapidly under the same broad scaling forces commonly associated with Moore's law. In magnetic storage, this trend is often described as Kryder's law: hard-

disk areal density historically increased exponentially, moving commercial systems from megabytes and gigabytes to terabytes, petabytes, and eventually hyperscale exabyte deployments. That progress was remarkable, but it was not achieved by simply making the same devices larger. Each generation required new device physics, firmware, and storage-management algorithms to maintain reliability, latency, endurance, and cost. Ex. 1017 (‘Kryder’s Law’).

26. For hard disk drives, manufacturers introduced technologies such as shingled magnetic recording (SMR), which increases areal density by overlapping adjacent tracks, and heat-assisted magnetic recording (HAMR), which uses localized heating to write smaller, more stable magnetic bits. For flash storage, manufacturers increased bits per cell from SLC to MLC, TLC, and QLC, and moved from planar NAND to 3D NAND. These techniques increased capacity and lowered cost per bit, but they also made device behavior more complex and workload dependent. Ex. 1018 (‘Shingled Magnetic Recording’); Ex. 1019 (‘Addressing Shingled Magnetic Recording Drives with Linear Tape File System’); Ex. 1020 (‘SMR: The Next Generation of Storage Technology’); Ex. 1021 (‘Lowering HAMR Near-Field Transducer Temperature via Inverse Electromagnetic Design’).

27. Flash storage or memory refers to persistent memory with no moving parts invented in the 1980s based on EEPROM technology. There are two primary types of flash memory, NOR Flash and NAND Flash. NOR uses parallel

connections for fast random access, while NAND uses series connections for high density. This requires NAND memory data to be written in large blocks rather than individually which led to cheaper, higher-capacity storage, making solid-state drives which included NAND memory more popular in the 1990s through 2000s.

B. Scaling Introduced New Device-Level Constraints

28. SMR illustrates the tradeoff. By writing overlapping tracks, SMR increases density but limits fine-grained in-place updates. In host-managed or host-aware SMR, zones are subject to sequential-write constraints; modifying data within a zone may require rewriting a larger band or zone. Drive-managed SMR can hide this behavior behind a conventional block interface, but doing so requires internal buffering, mapping, caching, and cleaning policies that affect performance, especially under random-write workloads. Ex. 1018 (‘Shingled Magnetic Recording’); Ex. 1022 (‘Read Disturb Errors in MLC Nand Flash Memory: Characterization, Mitigation, and Recovery’).

29. NAND flash scaling created different constraints. Multi-level NAND stores multiple bits per cell by assigning values to threshold-voltage ranges; as the number of programmed states increases and process geometry shrinks, voltage margins narrow. By the mid-2010s, MLC and TLC NAND were understood to present endurance, retention, read-disturb, and read-threshold tradeoffs that SSD controllers had to manage. The tighter voltage margins generally reduce

program/erase endurance, increase the need for error correction and read-retry mechanisms, and can increase latency. Controllers respond with wear leveling, garbage collection, bad-block management, read-retry calibration, SLC caching, and stronger error correction. Ex. 1020 (‘SMR: The Next Generation of Storage Technology’); Ex. 1021 (‘Lowering HAMR Near-Field Transducer Temperature via Inverse Electromagnetic Design’).

30. These technologies involve explicit tradeoffs. SLC, MLC, and TLC differ in capacity, cost, latency, and endurance. Over-provisioning provides another example: reserving more physical flash for controller-managed spare area can improve garbage collection, reduce write amplification, and increase endurance and sustained write performance, but it reduces user-visible capacity. Lower over-provisioning increases usable capacity and lowers cost per usable gigabyte, but can degrade performance and endurance as the drive fills. Ex. 1023 (‘Design Tradeoffs for SSD Performance’).

C. Traditional Interfaces Hid Device Complexity

31. A primary drawback of NAND memory is that each memory cell can only endure a finite number of write and erase cycles before it wears out. Manufacturers and designers address this problem with different approaches. For example, wear-leveling could be used to evenly distribute write and erase cycles across all NAND memory cells. As another example, systems could employ a defect

management or bad block management policy that accounted for defective memory cells.

32. Traditionally, device manufacturers hid this complexity behind a simple logical block interface. A host issued reads and writes; the device firmware decided how to translate logical addresses to physical media, buffer small writes, perform garbage collection, distribute wear, avoid defective blocks, and recover from media errors. If a hardware technology favored large contiguous writes, the device could collect many small writes and destage them as a larger internal operation. If Flash required block erasure before page rewriting, the flash translation layer handled the copying and indirection. This abstraction preserved compatibility and simplified software, but it prevented sophisticated users from directly exploiting knowledge of device behavior.

D. Hyperscale Systems Drove Software-Programmable Policies

33. The rise of hyperscale operators changed the economics. Companies such as Google and Meta operate storage at such scale that small inefficiencies in capacity utilization, write amplification, power, or tail latency can have large total-cost-of-ownership effects. They also control much of the stack, including applications, distributed storage software, operating systems, and hardware procurement. That made hardware/software co-optimization practical: rather than

hiding all device characteristics, storage systems could expose selected characteristics to software and let software choose policies suited to the workload.

34. This design philosophy was established well before November 2016. Google's File System was designed around Google's application workloads and technological environment, rather than traditional file-system assumptions, and later discussions emphasized that Google could adjust both the file system and the applications above it. HDFS similarly targets large files, streaming access, high aggregate bandwidth, and fault tolerance on commodity hardware. Ceph uses policy-driven placement through CRUSH so clients and storage daemons can compute where objects and replicas should be placed without centralized lookup tables. Ex. 1024 ('Understanding Intrinsic Characteristics and System Implications of Flash Memory Based Solid State Drive'); Ex. 1025 ('The Google File System'); Ex. 1026 ('GFS: Evolution on Fast-Forward'); Ex. 1027 ('The Hadoop Distributed File System'); Ex. 1028 ('Ceph: A Scalable, High-Performance Distributed File System').

E. Host-Programmable SSD Policies Were Known Before November 2016

35. Policy-based storage management long predated the '023 patent. IBM implemented hierarchical storage management for MVS in 1978 and later introduced DFSMS, a policy-based framework using administrator-defined classes and groups for allocation, performance, availability, migration, backup, retention, and storage

groups. IBM Spectrum Scale likewise documented pre-2016 information lifecycle management policies for file placement, migration, deletion, listing, and external storage pools. Ex. 1029 ('CRUSH: Controlled, Scalable, Decentralized Placement of Replicated Data'); Ex. 1030 ('Two Decades of Policy-Based Storage Management for the IBM Mainframe Computer'); Ex. 1031 ('DFSMSHsm Primer'); Ex. 1032 ('GPFS: Advanced Administration Guide'); Ex. 1033 ('IBM Spectrum Scale (formerly GPFS)').

36. Hierarchical or tiered storage systems apply policies that migrate and store data based on certain characteristics of the data. For example, a policy could store data in different tiers, storing hot data (frequently used data) in the more expensive disk drives in one tier, while cold data (less frequently used or LFU data) in cheaper, long-term tape archives in another tier.

37. Device-level interfaces evolved in the same direction. NVMe 1.0 and 1.1 provided a high-performance host-controller interface with host-configurable features that affected device behavior. Open-Channel SSDs went further: the April 2016 Open-Channel SSD NVMe Specification exposed physical media geometry and internal parallelism so host software could implement flash-management policies such as data placement, bad-block handling, garbage collection, and wear leveling. Ex. 1034 ('NVM Express Revision 1.0'); Ex. 1035 (NVM Express

Revision 1.2’); Ex. 1036 (‘NVMe Specification Revision 1.2’); Ex. 1037 (‘Storage Security Subsystem Class: Opal Specification Version 2.00’).

38. SCSI/SBC provides the same point in a narrower form. Before the priority date, SBC-3 WRITE SAME and UNMAP commands let the host send command fields or directives that told the device to write a pattern, apply protection-related treatment, or deallocate/unmap logical blocks, with device firmware carrying out the resulting media operations. Ex. 1038 (‘SCSI Block Commands - 3 (SBC-3)’); Ex. 1039 (‘SBC-3 Thin Provisioning Commands’); Ex. 1040 (SCSI Block Commands - 4 (SBC-4’).

F. More-Complex Policies Used Request Attributes and Data-Processing Rules

39. RAID can process content for storage by mirroring content, striping data across multiple disks, and calculating parity information so that, if a disk fails, the missing content can be reconstructed from the surviving data and parity. Ex. 1051 (‘Understanding Hard Drive Media Defects White Paper – Servers’). The original RAID paper and later survey described how disk arrays used striping and redundancy to improve performance and reliability. Ex. 1041 (‘A Case for Redundant Arrays of Inexpensive Disks (RAID)’); Ex. 1042 (‘RAID: High-Performance, Reliable Secondary Storage’). Commercial storage products made RAID policy choices concrete in controller software. Pure Storage's FlashArray implemented RAID policies in the storage controller to protect against device

failures, bit errors, and performance variability while EMC's XtremIO used row-and-diagonal parity, and distributed free space rebuilding policies to improve data reliability. Ex. 1043 ('Purity: Building Fast, Highly-Available Enterprise Flash from Commodity Components'); Ex. 1044 ('XtremIO Data Protection (XDP): Flash-Specific Data Protection').

40. Pure Storage RAID-3D and EMC XtremIO XDP are examples of complex configurable storage policies implemented by storage systems: controller software selected and implemented policies for redundancy level, parity layout, spare-space use, rebuild behavior, fault tolerance, performance, and flash endurance. Other pre-priority publications also disclosed storage requests carrying policy-like criteria, metadata, or attributes for replication, placement, searchability, storage-server policy engines, cloud-storage routing, and rule-based file operations. Ex. 1005 (Olster); Ex. 1045 (Armitano); Ex. 1046 (Devarakonda); Ex. 1047 (Vijayakumar); Ex. 1048 (Gollub); Ex. 1049 (Puranik); Ex. 1050 (Hardy).

G. Policy-Programmable Storage at Multiple Layers

41. By November 2016, storage systems were policy-programmable at virtually all layers. At the cluster level, distributed storage software could choose replication, erasure coding, placement, failure domains, rebalancing, and migration. At the server or storage-head level, file systems, object stores, and key-value stores could route requests to particular pools or devices. At the device level, controllers

could implement software-configurable policies for caching, error recovery, wear leveling, garbage collection, over-provisioning, bad-block avoidance, shingled-zone management, and data placement.

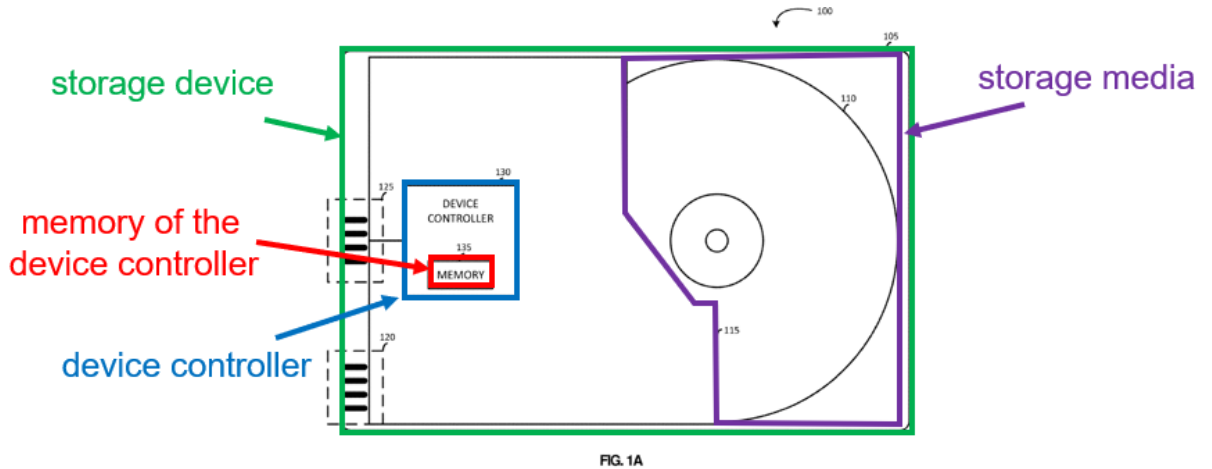
42. This layered policy architecture enabled unprecedented scale while preserving fault tolerance and performance. A system could apply one policy to write-heavy data and another to read-mostly data; one policy to temporary data and another to durable content; one policy to latency-sensitive data and another to archival data; or one policy to fast SLC-like regions and another to TLC/QLC bulk storage. The engineering overhead increased, but software-programmable policies enabled hardware/software co-optimization and total-cost-of-ownership improvements that were especially important at hyperscale. Against that background, storage device policies that are received, updated, and applied by a device controller were part of a well-developed progression from policy-based storage management to distributed placement policies, host-configurable device features, and software-defined flash. *See, e.g.*, Ex. 1005 (Olster); Ex. 1006 (Hashimoto); Ex. 1007 (Kilian); Ex. 1008 (Davis); Ex. 1009 (Eleftheriou); Ex. 1010 (Meir); Ex. 1011 (Salessi), Ex. 1015 (Lovinger).

V. THE '023 PATENT

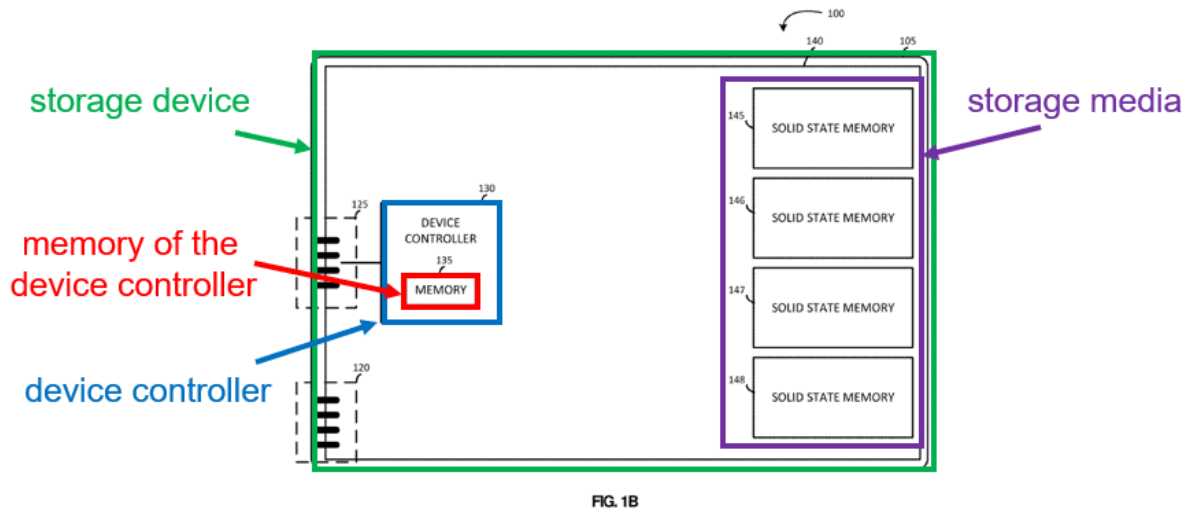
A. Overview

43. The '023 patent generally relates to “data storage devices with configurable policy-based read and write behavior.” Ex. 1001, 1:18-20. The '023 patent explains that “data storage devices come in a wide variety of technologies and configurations such as solid-state drives, hard disk drives, floppy disk drives, and the like” but that “[e]ach of these devices (DRAM, NAND, magnetic disks) has imperfections.” *Id.*, 1:24-33. “[T]o deal with these imperfections, and still have a functional and performant device, all of these devices have trade-off decisions made that affect storage space, reliability, speed, etc.” *Id.*, 1:33-36.

44. But the '023 patent identifies the problem at the time as this: “These trade-off decisions have largely been made by the manufacturer at the time of design and/or manufacture” and provide users “very little to no influence” on how data is stored. *Id.*, 1:37-41. The '023 patent attempts to address this problem using a routine storage device policy that allows “a user to customize the operation of the storage device.” *Id.*, 3:13-15.



Ex. 1001, Fig. 1A (annotations in color)



Ex. 1001, Fig. 1B (annotations in color)

45. The '023 patent describes **storage devices** include a **device controller** and **storage media**, such as hard disk drives (Figure 1A) or solid state memory (Figure 1B). *Id.*, FIGS. 1A, 1B, 1:45-46. The '023 patent describes that the device controller 130 can receive a storage device policy from a host, allowing the policy to be user-configurable. “[T]he device controller 130 uses a storage device policy...to determine where to store the content.” *Id.*, 14:42-44. The '023 patent

explains that “[t]he determination can relate partially to the content being stored, and partially to the storage media 110.” *Id.*, 14:44-46.

46. The '023 patent provides examples of policy that indicate a recording method: “For example, if the content contains an indication of priority level, such as information indicating that certain content is highly critical, the device controller 130, based on the storage device policy, may determine that the content should be stored on the storage media 110 at a low density, so the likelihood of data corruption will be minimal.” *Id.*, 14:46-52. As another example, “the storage device policy may instruct the device controller 130 to avoid storing highly critical content in sectors that have defects.” *Id.*, 14:53-56.

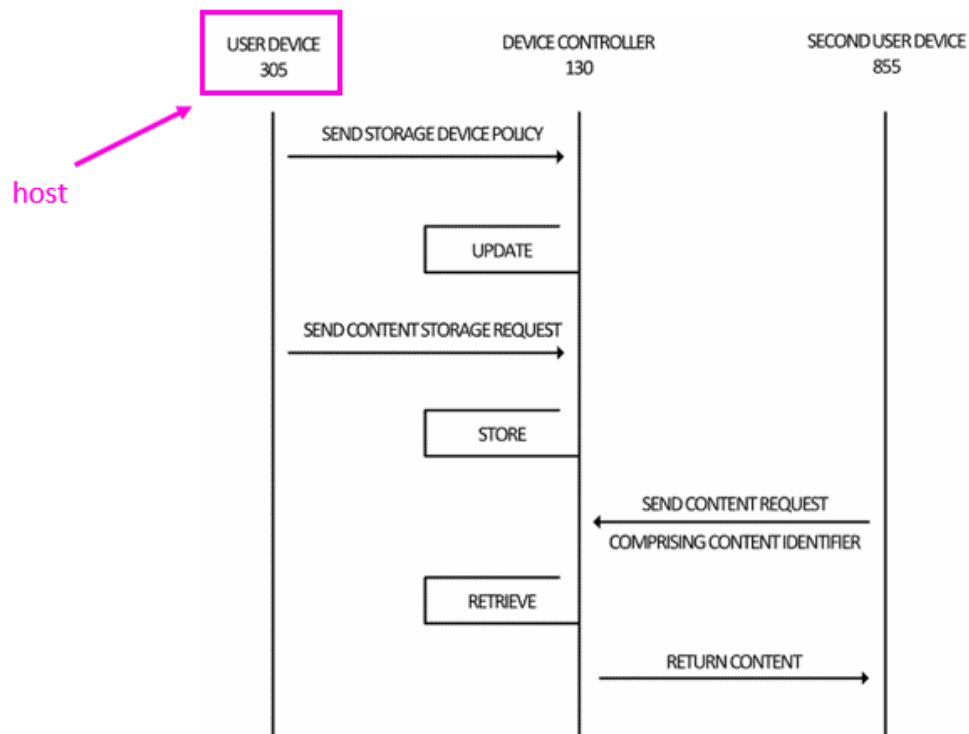
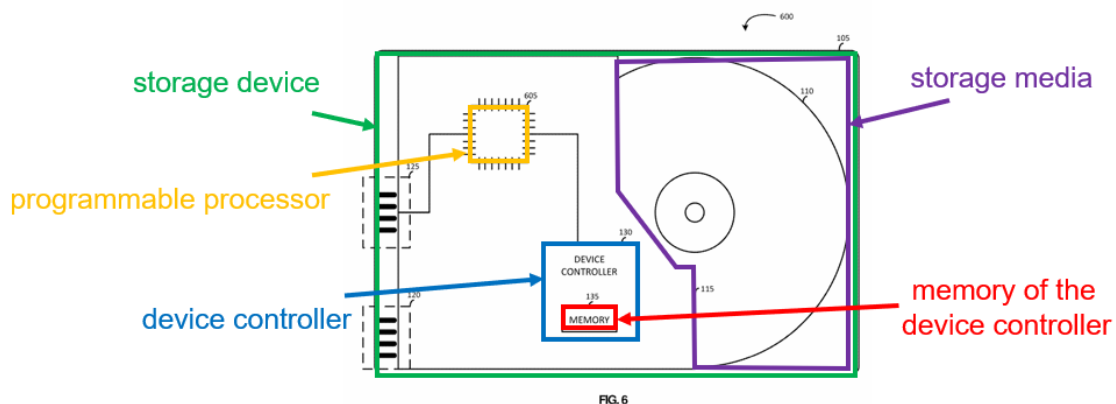


FIG. 9D

Ex. 1001, Fig. 9D (annotations in color)

47. In Figure 9D, the '023 patent describes one implementation of the operational sequence of this system of writing and reading content according to a storage device policy. *Id.*, FIG. 9D, 23:14-24:22. The device controller 130 receives a storage device policy from the host (here, **user device 305**). *Id.*, 23:15-17. Subsequently, the host “then sends a request to the device controller 130 to store content.” *Id.*, 23:33-35. Based on the storage device policy, the device controller determines “how to store the content” and “writes the content to the storage media.” *Id.*, 23:43-45, 23:52-53. The device controller 130 “can also store a record of the write (the storage information) in **memory 135**,” which “can include a unique identifier for the content.” *Id.*, 23:54-58. The device controller 130 can then receive “a content request” and “retrieves the requested content according to the storage device policy” and storage information. *Id.*, 23:64-65, 24:19-20, cls. 1, 9.



Ex. 1001, Fig. 6 (annotations in color)

48. In another embodiment, the **storage device** additionally includes a **programmable logic device 605**, or programmable processor, that “can be programmed by user of the drive 600 for any operation need that is desired.” *Id.*, 11:21-23. Accordingly, the programmable processor can implement the storage policy, but also other functions such as “custom encryption through hardware” or “compression.” *Id.*, 11:23-41. Code (or program instructions) stored on the **memory of the device controller** can then direct the device controller to store the processed content utilizing the recording method specified by the storage device policy. *Id.*, 11:61-63.

49. The '023 Patent also describes another embodiment in which a write-protection policy prevents data from being deleted upon a delete request. As described in reference to Figure 9C, “[t]he device controller 130 uses the storage device policy to determine...how to store the content. In an implementation, this determination includes a determination that the content should be stored as write-protected (or write-once) such that it will not be deleted later or overwritten.” *Id.*, 22:29-43.

B. Prosecution History

50. On May 3, 2019, the Examiner issued a non-final rejection of all pending claims on both §§ 102 and 103 grounds over Olster. The Examiner found Olster disclosed the “replication and placement component of a policy defines the

replication and placement of data objects in the data storage system” anticipating the independent claims, and that the “remaining dependent claim features are expressly or inherently found in the applied prior art.” Ex. 1002, 122-126. The Examiner also found the dependent claims obvious, stating each of “[t]he dependent claim features, while part of the invention, appear to be well known and their relevance not essential to the main invention found in the independent claim(s).” Ex. 1002, 127.

51. On October 3, 2019, the Applicant amended the claims to recite “at least one storage device policy [that] indicates at least a storage density.” Ex. 1002, 163-64. On October 5, 2019, less than a month later, the Examiner issued a final rejection of all claims on § 103 grounds over Olster in view of US 2002/0036852 to Leow et al. (a publication assigned to Seagate Technology LLC). *Id.*, 172-178. Notably, the Examiner stated “storing or recording data in accordance with densities to manage errors/corruption/error rate on a disk appears to be well known as found in Leow.” *Id.*, 178.

52. On April 24, 2020, the Applicant removed all references to storage density in the claims, and amended them to require that the controller receives “from a host” a storage policy that “indicates at least a recording method for writing content to the storage media.” *Id.*, 224. The Applicant stated that Olster only describes “a searchable and distributed database” including “a controller for a storage node which distributes storage policies and handles search queries.” *Id.*, 233 (emphasis added).

I do not agree with this characterization. For example, the Applicant stated “*Olster* does not contemplate a storage controller or storage system which stores content on a storage media by utilizing a recording method, received from a host, for writing content to a storage media.” *Id.* But as I discuss below, *Olster* is not just a searchable and distributed database system, but also a data storage system that includes storage nodes with controllers that store data in the storage media based on a policy received from a host that indicates the writing method used to stored data, such as a redundancy scheme. *See also* Ex. 1005, [0030] (“[t]he policies may be stored on and distributed from a client 102 [i.e., host] to the data storage system 100.”), [0031] (“defines the replication and placement of data objects in the ***data storage system***”).

53. On May 11, 2020, the Examiner issued a notice of allowance with no reasons for allowance. *Id.*, 239-45. The ’023 patent issued on September 15, 2020.

C. Priority Date

54. I have been asked to assume that the priority date of the ’023 patent is November 7, 2016. I have not been asked to determine whether the ’023 patent is entitled to that priority date.

D. Person of Ordinary Skill in the Art

55. A person of ordinary skill in the art in 2016 would have had at least a bachelor’s degree in electrical engineering, computer science, computer engineering, or a similar technical field, with at least two years of experience in the

field of data storage systems. Additional experience could substitute for less education, and additional education could likewise substitute for less experience.

56. My opinions do not turn on this precise definition, and the challenged claim would be unpatentable from the perspective of any reasonable person of ordinary skill in the art at the relevant time.

57. I believe that I would have qualified as a person of at least ordinary skill in the art as of the earliest alleged November 7, 2016 priority date of the '023 patent. I believe that I have sufficient level of knowledge, experience, and education to provide an expert opinion in the field of the '023 patent.

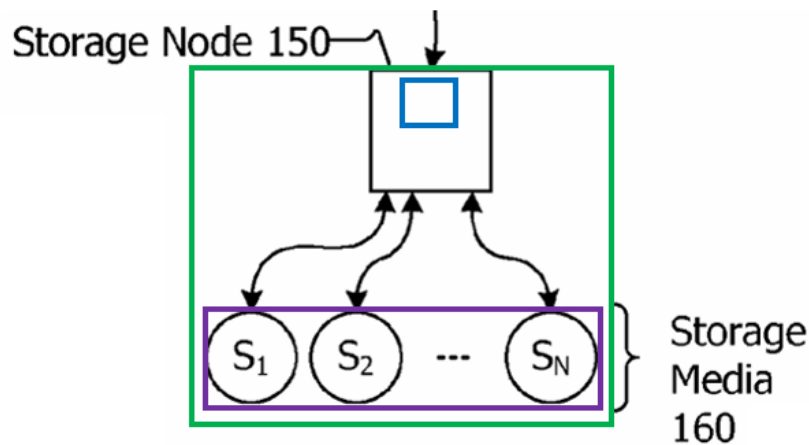
VI. CLAIM CONSTRUCTION

58. I have applied the plain meaning of the terms and phrases in the claims in my analysis, as a person of ordinary skill in the art would have understood those terms and phrases at the relevant time. I understand that the Board will construe the claims using the same claim construction standard that would be used in district courts. My declaration will establish that the prior art meets each of the claim limitations under its plain meaning. I have not been asked to consider any express constructions.

VII. THE PRIOR ART

A. Olster (Ex. 1005)

59. Olster describes a searchable data storage system that includes storage nodes that are made up of storage media and a controller, comprising a processor and memory. Ex. 1005, Abstract.

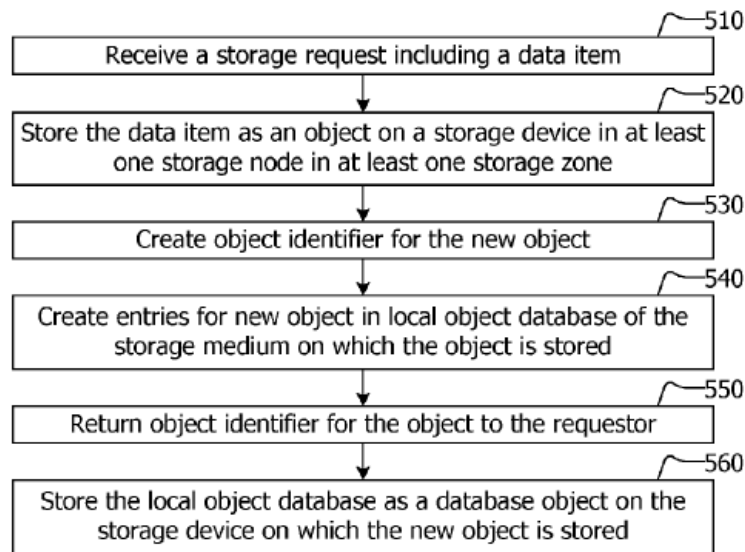


Ex. 1005, Fig. 2 (excerpted and with annotations in color)

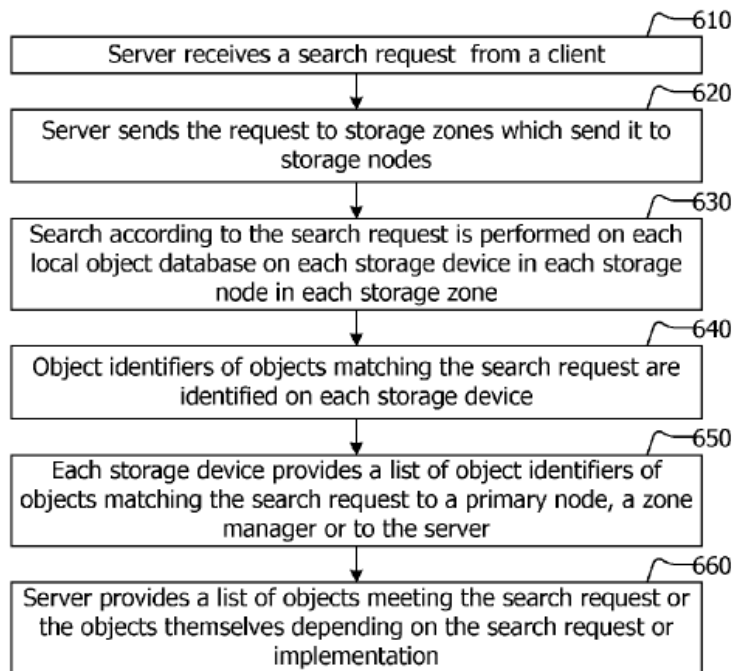
60. Olster teaches a storage node 150 that includes a controller and storage media, such as hard disk drives or flash memory. Ex. 1005, FIG. 2, cl. 1 (“each node comprises a plurality of storage devices and a controller, the controller including a processor and memory”), [0024] (“The controller may communicate with and allow for communication with the storage media”), [0022] (“The term ‘storage media’ is used herein to refer to any configuration of hard disk drives (HDDs), solid-states drives (SSDs),...or other similar storage media”), [0029]

(“Storage media include magnetic media such as hard disks and tape, flash memory”).

61. Olster explains that the controller can receive a storage policy from a **host**, allowing the policy to be user-configurable. *Id.*, [0030] (“The policies may be...distributed from a client 102 to the data storage system 100...and to all nodes in the data storage system.”), [0046] (“In another implementation, the specific distribution instructions...may be hard coded or system defined or *earlier set by a system administrator*.”). “[F]iles and other data may be partitioned into smaller portions and stored as multiple objects among multiple storage media 160 in a storage node 150...according to [a] storage policy.” *Id.*, [0030]. Olster explains that the policy can define the “placement of data objects in the data storage system,” *i.e.*, where the data is stored. *Id.*, [0031]. For example, the policy “may include specific distribution instructions for the data item,” which may “specify the number of zones in which the data item should be stored, the number of nodes in which the data item should be stored, a specific redundancy scheme identifier, and the like.” Ex. 1005, [0046].



Ex. 1005, Fig. 5



Ex. 1005, Fig. 6

62. Olster describes a method of writing and reading data according to a storage policy. *Id.*, FIGS. 5 (storage request), 6 (content request), [0046]-[0051] (“a

storage request...may specify the storage policy for the data item”), [0052]-[0054] (“The server then provides...the objects themselves depending on the search request.”). “[T]he system stores the data item as an object according to a policy (representing specific distribution instructions...) provided with the request or [] included in the system such that the data item is stored on at least one storage device.” *Id.*, [0047]. “The system creates an object identifier for the new object” that “may include a group identified (‘group ID’) and an index.” *Id.*, [0047], [0041]. “The group ID may be used to map the object to a particular...storage device” and “the system stores a portion of mapping information in memory.” *Id.*, [0041].

B. Davis (Ex. 1008)

63. To the extent that Olster does not disclose a storage policy that *indicates at least a recording method for writing content to the storage media*, this would have been obvious in view of Davis.

64. Davis describes a policy for writing data to flash memory based partially on the content being stored, and partially on the storage media. Ex. 1008, Abstract. “The method includes determining at least one property of a data and determining to which type of a plurality of types of flash memory to write the data, based on the at least one property of the data.” *Id.* Davis teaches “This policy...mov[es] data from the type of flash memory having the higher number of bits per cell to the type of flash memory having the lower number of bits per cell in

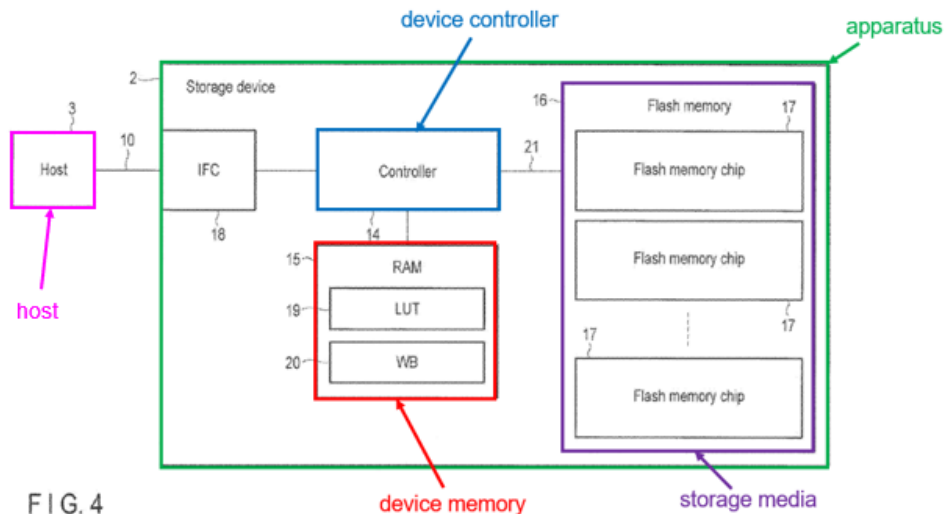
response to a data residence time of the data exceeding a residence time threshold value.” *Id.*, 11:11-15. This means, for example, that hot data (data that is frequently used) that has priority may be stored on flash memory having a lower number of bits per cell to minimize the likelihood of data corruption. *See also id.*, 11:7-11 (“One policy is moving data from a type of flash memory having a lower retention time to a type of flash memory having a higher retention time in response to a data residence time of the data exceeding a residence time threshold value.”).

65. Davis also teaches, as another example, “[o]ne policy is writing text data to a type of flash memory having a lower read error rate and writing image data to a type of flash memory having a higher read error rate.” *Id.*, 9:37-39. Thus, this policy instructs the controller to avoid storing critical data more sensitive to corruption, such as text data, in memory with more potential defects. *See also id.*, 9:39-45.

66. “For example, if the content contains an indication of priority level, such as information indicating that certain content is highly critical, the device controller 130, based on the storage device policy, may determine that the content should be stored on the storage media 110 at a low density, so the likelihood of data corruption will be minimal.” *Id.*, 14:46-52. As another example, “the storage device policy may instruct the device controller 130 to avoid storing highly critical content in sectors that have defects.” *Id.*, 14:53-56.

C. Hashimoto (Ex. 1006)

67. Hashimoto describes a storage device that includes **controller 14** (*device controller*), **RAM 15** (*device memory*), and **flash memory** (*storage media*). The **host 3** provides commands, which are transmitted to the storage device. Ex. 1006, [0121], Fig. 4. These commands include tier attributes (*storage device policy*) which determine the recording method to be used for certain data. *Id.*, [0193]-[0195].



Ex. 1006, Fig. 4 (annotations in color)

68. “For example, when data is required to have high reliability, the host 3 may designate a tier attribute giving a higher priority to the data reliability than to the read speed.” *Id.*, [0202]. Alternatively, “when data is required to have a high read speed rather than high reliability...the host 3 may designate the tier attribute giving a higher priority to the read speed than to the data reliability.” *Id.*

69. Hashimoto further discusses LBAs (logical addresses) that are used to identify specific content (content identifiers). Specifically, “the controller 14 updates the lookup table (LUT) 19 to map the LBAs corresponding to the written data to the physical addresses corresponding to physical storage locations at which the data is written.” *Id.*, [0121]. This allows the system to retrieve data at a particular location that is unique to that data, based on the LBA.

D. Kilian (Ex. 1007)

70. Kilian is directed to “a method of processing data in a computer system comprising at least one host and at least one storage system.” Ex. 1007, 1:56-58. The method itself comprises a controller receiving a request to delete data in the system and determining, based on a previously-defined retention period for that data, whether the delete request can be fulfilled. *Id.*, 1:58-64. If the controller determines that the previously-defined retention period for the data has not yet lapsed, the controller will deny the delete request; alternatively, if the retention period has lapsed, the controller will fulfill the request and delete the data at-issue.

VIII. OPINIONS ON GROUND 1: CLAIMS 1, 7, 9, 14, AND 15 ARE OBVIOUS IN VIEW OF OLSTER

71. In my opinion, Olster renders obvious claims 1, 7, 9, 14, and 15. Olster describes a searchable data storage system that includes storage nodes that are made up of storage media and a controller, comprising a processor and memory. Ex. 1005, Abstract, [0024] (“The storage media 160 in a storage node 150 may be included in

a single cabinet, rack, shelf or blade...A controller may be included in the cabinet, rack, shelf blade with the storage devices...The controller may include a processor, volatile memory and non-volatile memory.”), [0025] (“The controller or primary node will also receive search queries from a server, distribute the search queries to all nodes in a storage zone, store the results of search queries as an object, and provide the results of the search to the requesting server.”), cls. 1, 10.

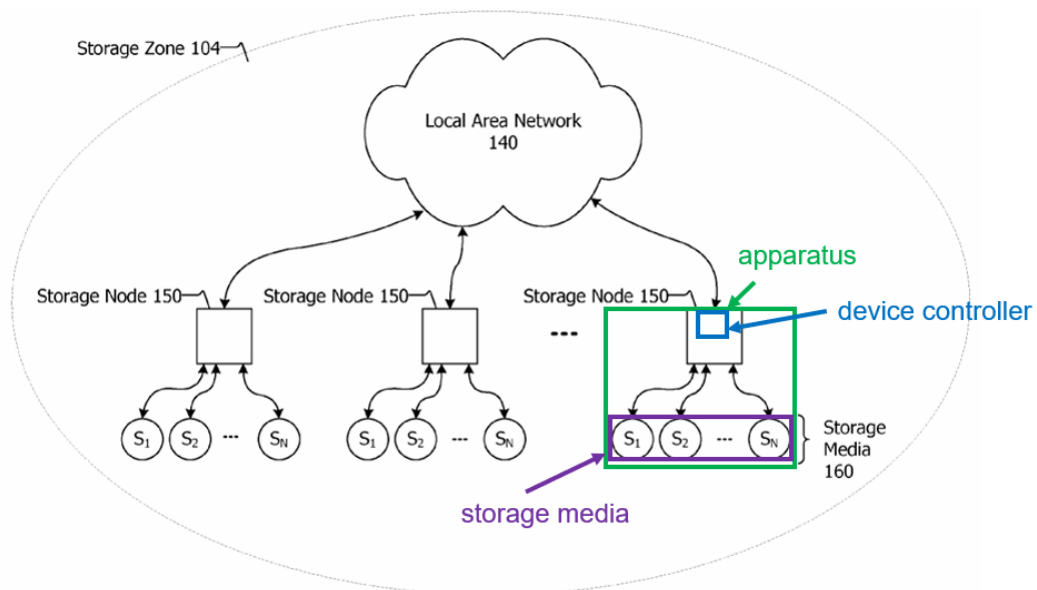


FIG. 2

Ex. 1005, Fig. 2 (annotations in color)

72. Olster also describes storage policy. Ex. 1005, [0030]-[0032] (“The object including its constituent data portion may be stored among storage nodes and storage zones according to the storage policy specified by the storage policy identifier included in the object.”). These policies in Olster “have two components: [1] a replication and placement component and [2] a searchability component.” *Id.*,

[0031]. “The replication and placement component of a policy defines the replication and placement of data objects in the data storage system.” *Id.* “The searchability component of a policy may include a listing of metadata concerning objects that may be searchable.” *Id.*, [0032]. The data storage system reads and writes data called data objects or data items to “storage nodes...according to the storage policy.” *Id.*, [0030].

73. Olster discloses each limitation of claims 1, 7, 9, 14, and 15 as features in a single embodiment. I have been asked to assume that no motivation to combine or reasonable expectation of success is required when all features are disclosed in one embodiment as described in my analysis in Olster. Nevertheless, it is my opinion that a person of ordinary skill in the art would have been motivated to combine the disclosed features in Olster in a single system with a reasonable expectation of success.

74. The storage policy in Olster is not set at manufacture. Rather, Olster explains that the storage policy “may be...earlier set by a system administrator” or “distributed from a client 102.” *Id.*, [0046] (“the specific distribution instructions and/or the searchability of metadata may be hard coded or system defined or earlier set by a system administrator”), [0030] (“The policies may be stored on and distributed from a client 102 to the data storage system 100 and to all zones in the data storage system and to all nodes in the data storage system”). A person of

ordinary skill in the art would have known from Olster's teachings to allow an administrator or client to configure or program the storage policy as appropriate rather than limit policies to ones that were hard coded at manufacture. Allowing this configurability would allow the storage drive to be applied in more contexts, as the storage policy could be flexible. Further, an administrator or client would be able to adapt the data storage system to changing conditions.

75. For example, if the storage configuration changes, the administrator or client may choose to reconfigure the storage policy to be more suitable for the new storage configuration. In one set of circumstances, an SSD with only single-level cell (SLC) technology is swapped for an SSD that includes both SLC and multi-level cell (MLC) technology, which has a higher wear rate than SLC. It may be desirable to configure the policy so that the amount of writes to and reads from the SLC are higher than to and from the MLC. One way to implement this policy is to write more-frequently accessed data to the SLC and less-frequently accessed data to the MLC. It also may be desirable to update the policy based on the type of data or application being stored, if it does not align with the policy, which may have been configured for a different type of data or application. For example, if mission-critical data requires higher-performance memory access than what is feasible under the operative storage policy, the administrator or client may choose to fine tune the

policy such that the mission-critical data is stored in a way that improves throughput or reduces latency during reads or writes of that data.

76. A person of ordinary skill in the art would also have had a reasonable expectation of success because Olster teaches that distributing the storage policy through an administrator or client is an option for its data storage system. In particular, such a person would understand that storage policy is stored in the memory of the controller. Therefore, such a person would further understand that the policy can be updated through communications sent to the storage node, because that is where the policy is being stored. Such a system would have involved a combination of known technologies (data storage systems) according to known methods (communications) to yield the predictable result of a data storage system that implements a user-configurable storage policy. Therefore, a person of ordinary skill would have found a reasonable expectation of success in implementing Olster's disclosures as a system that teaches claims 1, 7, 9, 14, and 15.

A. Independent Claims 1, 9:

1. Preambles 1[pre]/9[pre]: “An apparatus comprising:”

77. I have been asked to assume the preambles of claims 1 and 9 are limiting. Under that assumption, it is my opinion that Olster discloses the preambles of claims 1 and 9, which recite “[a]n apparatus.”

78. Olster describes a storage node (*apparatus*). Ex. 1005, [0022] (“[In] FIG. 2, the storage zone 104 includes a plurality of storage nodes 150 which include a plurality of storage media 160.”); *see also* my discussion above in Section VII.A. Olster also describes that the various components that make up the storage node can be included in an enclosed space with other storage nodes or alone. *Id.*, [0024] (“The storage media 160 in a storage node 150 may be included in a single cabinet, rack, shelf or blade.”), [0027] (“The rack, shelf or cabinet containing a storage node 150 may include a communications interface that allows for connection to other storage nodes, a computing device and/or to a network.”). As I discuss for Limitations 1[a]/9[a]-1[b.viii]/9[b.vii], the storage node 150 includes the other limitations recited in the claim. *See* Sections VIII.2-11.

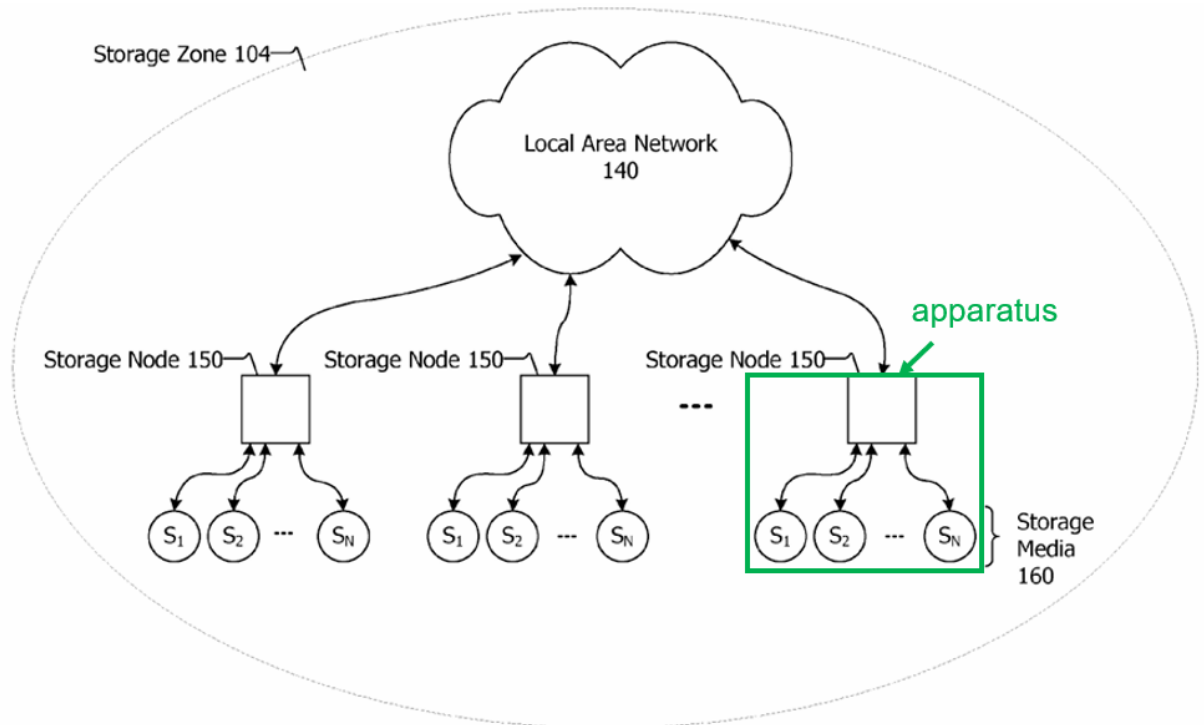


FIG. 2

Ex. 1005, Fig. 2 (annotations in color)

2. Limitations 1[a]/9[a]: “a storage media;”

79. In my opinion, Olster discloses 1[a]/9[a]: “a storage media.”

80. Olster describes *storage media* within the storage node 150 (*apparatus*). Ex. 1005, [0022] (“storage nodes 150 [] include a plurality of storage media 160”), FIG. 2 (I reproduced this figure below with annotations).

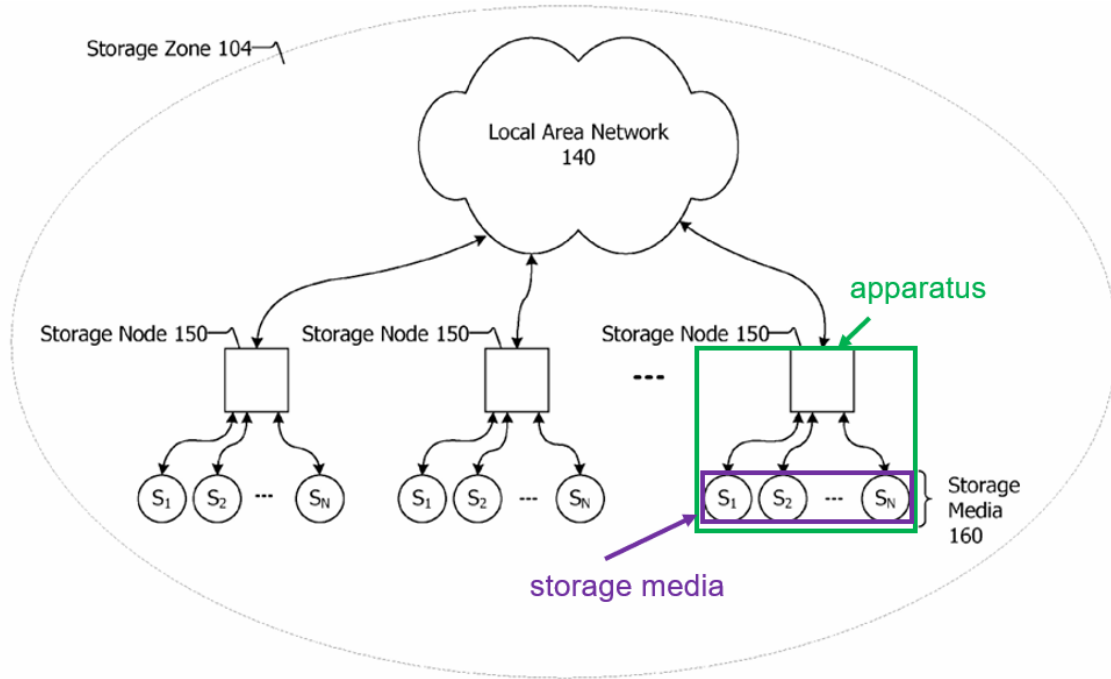


FIG. 2

Ex. 1005, Fig. 2 (annotations in color)

81. As a person of ordinary skill in the art would know, Olster explains “[s]torage media include magnetic media such as hard disks and tape, flash memory, and optical disks *Id.*, [0029]. Confirming this understanding, Olster also refers to storage media as storage device. *Id.*, Abstract (“The nodes include storage devices”), [0023] (“the physical size of the **media** in a storage node may differ, and/or...storage specification of the media in a storage node may not be uniform among all of the **storage devices** in the storage node.”), [0029] (“Storage devices include hard disk drives (HDDs) [and] solid-state drives (SSDs).”), cl. 1 (“each node comprises a plurality of storage devices and a controller”). For example, in discussing the storage media 160, Olster references back to the storage media by using storage device:

“When the *storage media* in a storage node are included in a single cabinet, rack, shelf or blade, they may be coupled with a backplane. A controller may be included in the cabinet, rack, shelf or blade with the *storage devices*.” *Id.*, [0024]. Therefore, Olster describes a storage node 150 that contains storage media 160 (*storage media*).

3. Limitations 1[b]/9[b]: “a device controller operatively coupled with the storage media configured to at least:”

82. In my opinion, Olster discloses 1[b]/9[b]: “a device controller operatively coupled with the storage media configured to at least.”

83. Olster describes “a controller for a node” (*device controller*). Ex. 1005, [0020], cl. 1 (“each node comprises a plurality of storage devices and a controller”), cl. 10 (“each storage node comprises a plurality of storage devices and a controller”).

84. The controller is operatively coupled with the storage media through a backplane that facilitates communication. *Id.*, [0024] (“When in a single cabinet or rack, storage nodes and/or constituent storage media may be coupled with a backplane.”), *id.*, (the controller “may be coupled with a backplane” and “may communicate with and allow for communications with the storage media”). Because both the storage media and the controller may be coupled to the backplane, the storage media and the controller are thus operatively coupled to each other.

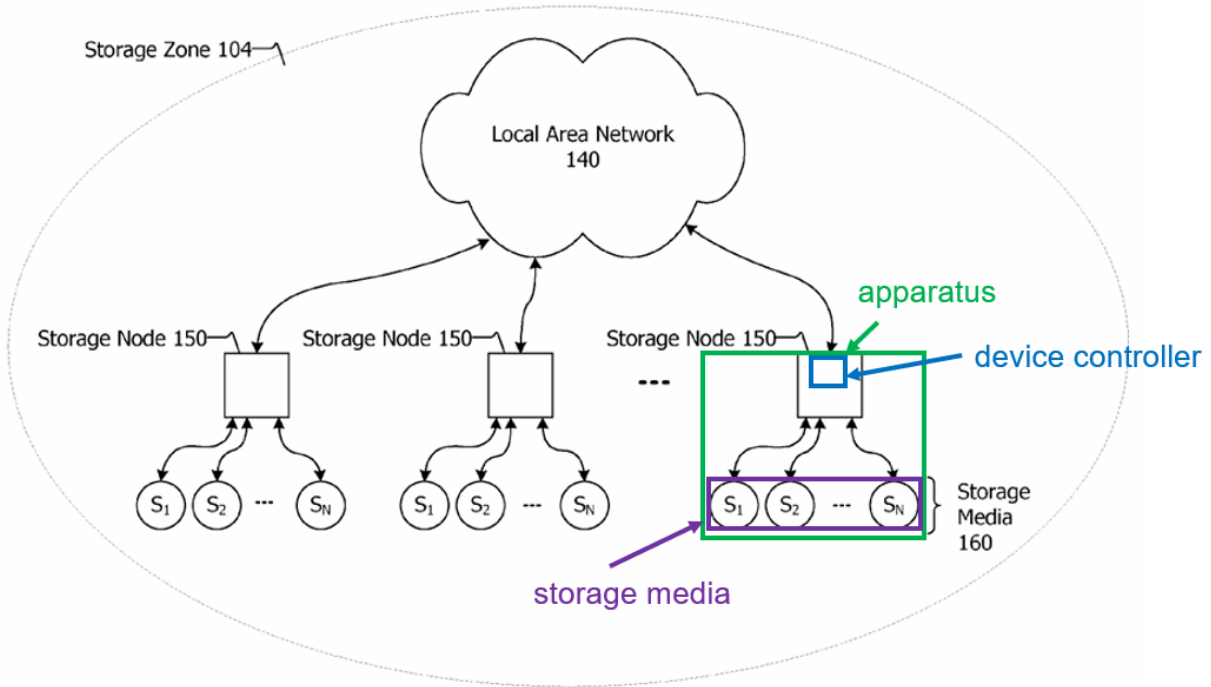


FIG. 2

Ex. 1005, Fig. 2 (annotations in color)

85. As I discuss for Limitations 1[b.i]/9[b.i]-1[b.viii]/9[b.vii], the controller of the storage node (*device controller*) performs storing and retrieving data objects (*content*) to and from storage media 160 (*storage media*) for those limitations. See Sections VIII.A.4-11.

4. Limitations 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media;”

86. In my opinion, Olster discloses 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media.”

87. Olster describes data for storage (*content*) referred to as the data portion of the data objects or data items. Ex. 1005, [0006] (“Various applications may store large numbers of documents, images, audio, videos, and other data as objects.”). Olster also described storage policies, as I discussed above (*see* Section VII.A). Further, Olster describes *at least one storage device policy* because Olster teaches that the storage system will store the data for storage (*content*) in the storage node, such as in SSDs or other storage devices, according to a storage device policy. *Id.*, [0030] (“[t]he object including its constituent data portion may be stored among storage nodes and storage zones according to the **storage policy**”) (emphasis added).

88. The described storage policies (*at least one storage device policy*) *indicates at least a recording method for writing content to the storage media* because they include a “replication...component” that “defines the replication...of data objects in the data storage system.” Ex. 1005, [0031]. “The replication...component may include specific distribution instructions for the data item.” *Id.*, [0046].

89. The replication component of the policy provides redundancy through a redundancy scheme such as “full replication”, “partial replication”, or a RAID policy. *Id.*, [0034]-[0035], [0046]. The replicas may be written before, after or during the write of data. *Id.*, [0034]. Olster discloses that the replication policy includes a RAID configuration. *Id.*, [0035] (“resiliency may be provided by using

intra-zone techniques such as by a RAID or other configuration.”). The replication component may also include “a specific redundancy scheme identifier” (*id.*, [0046]), which a POSITA would have understood to include a RAID-level policy, such as RAID 0, RAID 1, RAID 5, and RAID 6, that can be used within a SSD with multiple types of storage media.

90. A redundancy scheme or policy, such as the replication component of Olster’s policy, is a *recording method for writing content to the storage media* because it provides rules for how the data is recorded in the storage media. A policy providing for full replication, partial replication, or RAID will provide rules and instructions on a writing method for storing the data in the storage media such as striping the data (dividing it into blocks) and mirroring the data (making a copy of the data to store a redundant copy in other memory units) that will be applied on a write operation to store the data and applied on a read operation to recompute or rebuild the data.

91. Redundancy schemes are a type of media defect management because the purpose of the policy is to manage and address defects in the storage media. Ex. 1051 (IBM describing the advantages of using RAID for defect management). If a device fails, the redundancy policy will provide alternative ways to recover or rebuild the data.

92. The replication component of the policy of Olster manages defects in the storage media. Olster explains that when a redundancy scheme is applied, “access to data is possible when a zone goes down or is impaired or unreachable.” Ex. 1005, [0035]. A person of ordinary skill in the art would have understood that if a portion (e.g., block or page) of the SSD is defective and the data is unrecoverable, data recovery and rebuild is done through the RAID system or other redundancy scheme.

93. Olster also describes that the system receives the storage “policies.” *Id.*, [0046] (“[T]he system receives a storage request that...may include storage criteria....The storage criteria may, in different embodiments specify the storage policy....”). Because the controller is the part of the system that applies the policy, the controller of the storage node 160 (*device controller*) receives the policy to store the data objects according to the storage policy (*at least one storage device policy*). *Id.*, [0026] (“The controller may communicate with and allow for communications with the storage media.”), [0047] (“In one embodiment, the system stores the data item as an object according to a policy...”).

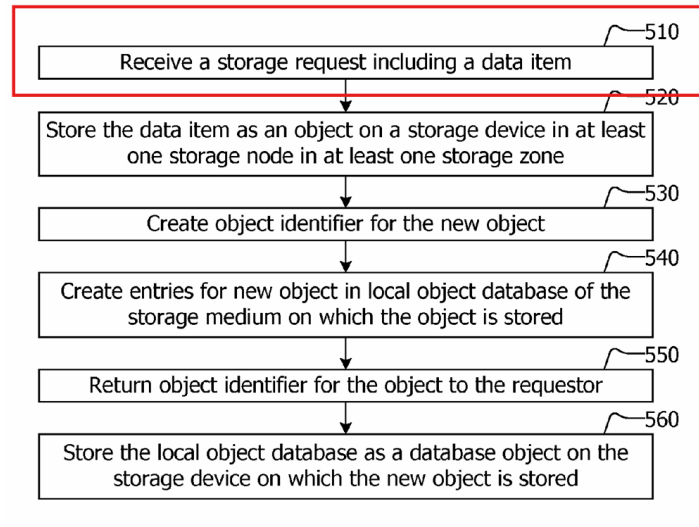
94. Olster also describes that these storage “policies” (*storage device policy*) “may be...distributed from a client 102,” including being “earlier set by a system administrator” (*receive, from a host*). *Id.*, [0030], [0046] (“[T]he system receives a storage request that...may include storage criteria...The storage criteria

may, in different embodiments specify the storage policy....”). Therefore, the client 102 or system administrator is acting as a *host* in providing the storage policies and storage criteria.

5. Limitation 1[b.ii]: “receive a storage request, from the host, to store content;”

95. In my opinion, Olster discloses 1[b.ii]: “receive a storage request, from the host, to store content.”

96. Olster describes a controller that “receives a storage request” (*receive a storage request*). Ex. 1005, [0046] (“To store or put a data item, the system receives a storage request that includes a data item (or reference or link to a data item) and may include storage criteria, and may include a set of metadata, as shown in block 510.”); FIG. 5. To the extent that Olster does not explicitly describe that it is the controller that is doing the receiving, this would have been obvious to a person of ordinary skill in the art. Olster elsewhere describes that it is “[t]he controller may communicate with and allow for communications with the storage media.” *Id.*, [0026]. Therefore, the receipt of a storage request for storage purposes would be a communication with the storage media that the controller would facilitate.



Ex. 1005, Fig. 5 (annotations in color)

97. The storage request that Olster describes is “received from a client application running on a client device” (*received, from a host*). *Id.* [0046]. Further, because the purpose of the storage request is to “store or put a data item” in the storage media, the request is to *store content*.

6. Limitations 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”

98. In my opinion, Olster discloses 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy.”

99. As I discussed above, Olster uses “storage media” and “storage device” interchangeably. *Id.*, [0029] (“Storage devices include hard disk drives (HDDs) [and] solid-state drives (SSDs).”); *see* Section VIII.A.2.

100. Olster describes that the controller stores a “data item,” (*content*) “as an object on a storage device” (*storage media*) “according to a policy (representing specific distribution instructions)” (*recording method indicated by storage device policy*). Ex. 1005, [0030] (“The object including its constituent data portion may be stored among storage nodes and storage zones according to the storage policy.”), [0047] (“In one embodiment, the system stores the data item as an object according to a policy (representing specific distribution instructions and a definition of which metadata fields are to be searchable.”), [0046] (“[t]he replication and placement component may include specific distribution instructions”).

101. Specifically, the controller will store the data item in a location indicated by the distribution instruction in the policy, which “may specify the number of zones in which the data item should be stored, the number of nodes in which the data items should be stored, a specific redundancy scheme identifier, and the like.” Ex. 1005, [0046]. As I discussed above for Limitation 1[b.i]/9[b.i], the client 102 or a system administrator controls the policy on how the data item is recorded. See Section VIII.A.4. In being able to place a particular data item in a particular zone with other copies, Olster’s data storage system can balance any competing concerns such as power consumption and cost that are typically inherent.

7. Limitations 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”

102. In my opinion, Olster discloses 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content.”

103. Olster describes that the controller includes memory (*memory of the device controller*). Ex. 1005, cls. 1 (“each node comprises a plurality of storage devices and a controller, the controller including a processor and memory”), 10 (“each storage node comprises a plurality of storage devices and a controller, each controller including a processor and memory”), [0024] (“The controller may include a processor, volatile memory and non-volatile memory.”).

104. Olster describes “object specific information including metadata,” which is *storage information for the content* because it includes information about how the object is stored. For example, the metadata will include the Location ID. Ex. 1005, [0040], Fig. 4; *see also* Ex. 1005, [0032] (“metadata is information associated with an object that describes attributes of the object”), [0039] (“Example metadata fields that may be stored in the object database as object specific information include a user identifier or ID for the author or owner or caretaker of the object, and pertinent time information for the object, including time created and time stored, and may include ATIME, CTIME and MTIME values for the object.”),

[0048] (“The database entries for the object are populated with object specific information or metadata according to the definition of the policy associated with the object.”), [0049] (“Additional object specific fields may also be stored as metadata regarding the object such as...the content of the data, such as, for example, name of person or location or subject, and/or [c] geo-location of where the data was obtained or captured.”).

105. Olster further describes that “objects...may be referenced by object identifier or OIDs” (*content identifier*). *Id.* [0040] (“The object identifier 400 includes a location identifier 404...[which] specifies a device, address, storage node or nodes where an object resides.”), [0049] (“the database entries for objects include an object identifier or OID”). A person of ordinary skill in the art would have understood that the object specific information or metadata (*storage information*) includes the OID (*content identifier*) because this information would provide specific details that would allow the content to be stored in a way that allows retrieval using the OID without the need of searching the object in the database based on its content. Ex. 1005, [0038]-[0039], [0053]

106. Olster expressly returns the OID after storage and explains that objects may be referenced by OID. Ex. 1005, [0041]. A person of ordinary skill in the art would have understood that a later request for that object could use the OID as the

field/value pair or object reference because the OID is the identifier Olster created and returned for that purpose.

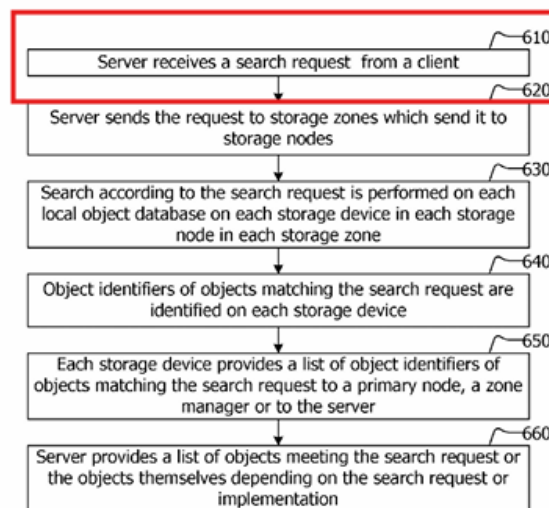
107. Olster explains that the OID may include a group identifier that “may be used to map the object to a particular storage node or storage device,” and that this mapping is found in the “mapping information” that is a part of the object specific information including the metadata (*storage information*). *Id.*, [0041]. The mapping information is object specific information because it includes information about the object such as its OID, which is also object specific information, as I discussed above. *Id.*, [0038]-[0039], [0041], [0053]. Olster further describes that “the system stores a portion of mapping information” which is part of the object specific information (*storage information*) “in memory” (*memory of the device controller*). *Id.*, [0041]. This can be compared to other types of storage in Olster, such as where “the storage nodes hold a portion of the mapping information in their memory.” *Id.*

108. A person of ordinary skill in the art would have understood that storing a portion of mapping information in the system’s memory is *record[ing] storage information . . . on a memory of the device controller*, because the mapping information provides detail on the destination of the content, which Olster describes is stored in the device controller (and specifically, in its memory).

8. Limitations 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content;”

109. In my opinion, Olster discloses 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content.”

110. Olster describes that the controller (*device controller*) *receives a* “search request from a client” (*content request*), as shown in block 610 below, that includes “at least one value [and] at least one field.” Ex. 1005, [0052], cls. 1, 10, FIG. 6.



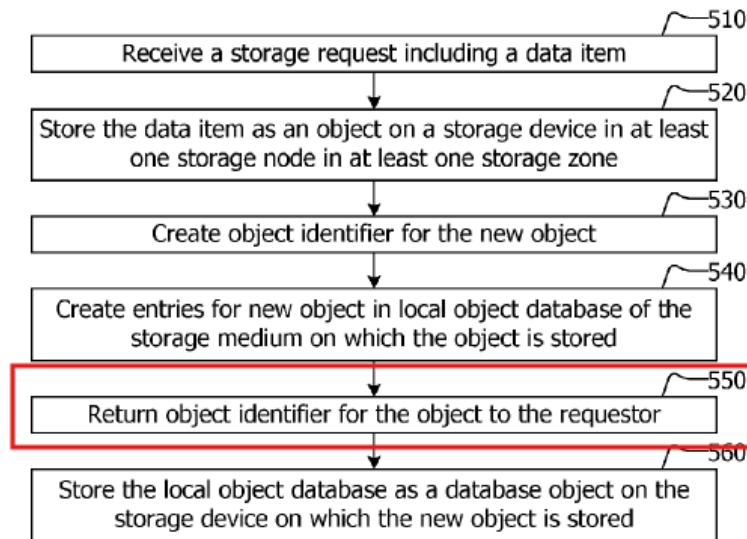
Ex. 1005, Fig. 6 (annotations in color)

111. The controller receives the request at step 610 because the controller is that part of the server (that is part of the storage node) that processes the requests.

Olster explains that “[e]ach of the storage nodes 150 may include one or more server computers.” Ex. 1005, [0022]. Server computers perform a service, and for Olster, it is to process the requests for the storage nodes.

112. Olster describes that “metadata fields [] may be stored in the object database as object specific information” and that “[t]he field or entries in the object database may be stored as field/value pairs.” *Id.*, [0039]. A person of ordinary skill in the art would have understood that the “at least one field [and] at least one value” are object specific information/metadata “field/value pairs” because they both correspond to the same object. *Id.*, [0052]. Such a person of ordinary skill would have further understood that one possible field/value pair is the OID (*content identifier*) because the OID is a type of object specific information/metadata that can determine the character and location of content. *See* Section VIII.A.7.

113. Olster describes that the OID was previously returned to the requester during execution of the storage request. “After the object is stored and the OID is created, the OID may be returned to the requester as shown in block 550.” Ex. 1005, [0051]. Therefore, after the object is stored (*subsequent to storing the content*), the controller receives, as part of the search request (*content request*), the OID (*content identifier*), as depicted below in FIG. 5.



Ex. 1005, Fig. 5 (annotations in color)

9. Limitations 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content;”

114. In my opinion, Olster discloses 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content.”

115. As I discussed above, Olster describes receiving a “search request” (*content request*) including the OID (*content identifier*). See Section VIII.A.8. As I also discussed above, Olster explains that the OID further includes a group ID “to map the object to a particular storage node or storage device.” Ex. 1005, [0041]; see Section VIII.A.7. Additionally, I discussed above that Olster describes that the mapping information derived from the group ID in the OID (*storage information for*

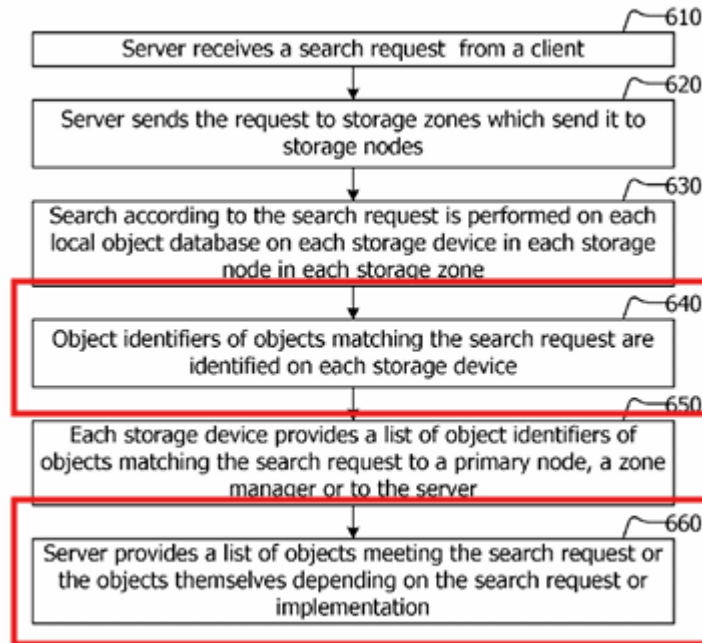
the content) is stored on the system memory (*memory of the controller device*). See Section VIII.A.7. A person of ordinary skill in the art would therefore have understood that *in response to the content request*, the controller *retrieves from* the system memory (*memory of the controller device*), the mapping information (*storage information for the content*) because the controller would need the OID, including the group ID, to locate the data item to be retrieved.

10. Limitations 1[b.vii]/9[b.vi]: “recover the content from the storage media, [at least in part according to the storage information]/[according to the storage information and according to the at least one storage device policy]”

116. In my opinion, Olster discloses 1[b.vii] (“recover the content from the storage media, at least in part according to the storage information”) and 9[b.vi] (“according to the storage information and according to the at least one storage device policy.”).

117. Olster describes storage device (*storage media*), as I discussed above. See Section VIII.A.2. Olster further describes that the storage device (*storage media*) stores the data objects (*content*). See Section VIII.A.6. Olster describes identifying on the “storage device” (*storage media*), “[o]bject identifiers of objects matching the search request . . . as shown in block 640” below (*storage information*). Ex. 1005, [0053], FIG. 6. “The server then provides a list of objects meeting the search request or the objects themselves,” (*content*) as depicted in block 650 below.

Id., [0054], FIG. 6. Thus, Olster describes *recovering the data objects (content) from the storage device (storage media)*.



Ex. 1005, Fig. 6 (annotations in color)

118. As I discussed above, Olster explains that the controller retrieves the object specific information including the mapping information (*storage information*) using the group ID portion of the OID (*content identifier*). See Section VIII.A.9. As I also discussed above, Olster explains that the mapping information (*storage information*) “map[s] the object to a particular storage node or storage device.” *Id.* A person of ordinary skill in the art would have understood that the controller *recovers* the data objects (*content*) from the storage device (*storage media*) at least

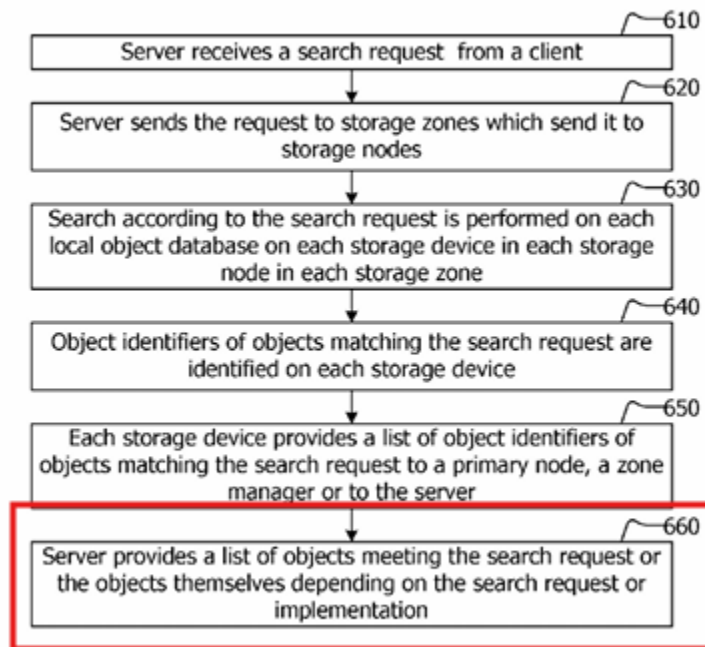
in part according to the mapping information (storage information) because looking up by the OID is a common approach that every storage device supports.

119. For claim 9 specifically, Olster also discloses that the content is recovered *according to the replication component of the policy (the at least one storage device policy)* because a person of ordinary skill in the art would have understood that the data would need to be recomputed or rebuilt as indicated by the redundancy policy. Ex. 1005, [0034]-[0035], [0052]. For example, for RAID-levels where striping is applied, the data will need to be rebuilt from the divided blocks. *See* Ex. 1041; Ex. 1042; Ex. 1051.

11. Limitations 1[b.viii]/9[b.vii]: “respond to the content request at least by returning the content to the host.”

120. In my opinion, Olster discloses 1[b.viii]/9[b.vii]: “respond to the content request at least by returning the content to the host.”

121. In FIG. 6, Olster depicts a “block 660” where the controller can provide “the objects themselves.” Ex. 1005, FIG. 6, [0054]. As I discussed above, Olster explains that step 610 of FIG. 6 includes *receiving a search request (content request) from the host. See* Section VIII.A.8. A person of ordinary skill in the art would have understood that the controller *responds to the search request (content request) at least by returning the objects (content) to the host* because “a list of objects meeting the search request or the objects themselves depending on the search request or implementation” are provided. Ex. 1005, FIG. 6.



Ex. 1005, Fig. 6 (annotations in color)

B. Dependent Claims 7, 14, 15:

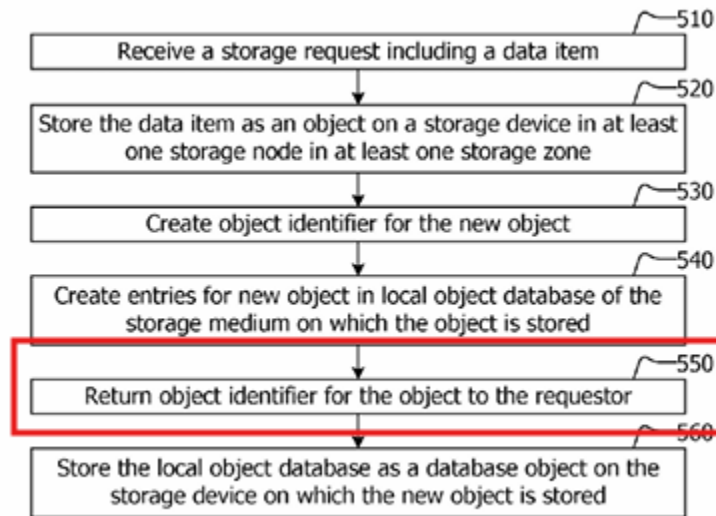
- 1. Claim 7: “The apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”**

122. Olster discloses claim 7: “[t]he apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

123. Olster describes an “object identifier” or “OID” (*content identifier*) that is used to identify the object “on a particular medium.” Ex. 1005, [0051]. Olster explains that the unique identifier portion of the OID (*content identifier*) may be “assigned by the controller” (*device controller*). Ex. 1005, [0042]; *see also id.*

[0040] (“The object identifier 400 includes a location identifier 404 and a unique identifier 406.”). The way that the “OID” is assigned can differ by system, but the assignment is critical to the proper storing and retrieval of content.

124. Olster explains that the “OID” (*content identifier*) is “returned to the requester.” *Id.*, [0051] (“After the object is stored and the OID is created, the OID may be returned to the requester, as shown in block 550.”), FIG. 5. Olster describes returning the OID (*content identifier*) to the requester (block 550) after receiving the storage request (block 510). *Id.*, [0046] (“The storage request may be received from a client application running on a client device.”), [0051] (“After the object is stored and the OID is created, the OID may be returned to the requester, as shown in block 550.”), FIG. 5. This would allow the requester to confirm that the system has retrieved the correct OID, and thus, content. Therefore, Olster describes that the OID (*content identifier*) is returned in a response to the storage request. *Id.*



Ex. 1005, Fig. 5 (annotations in color)

2. Claim 14: “The apparatus of claim 9, wherein the storage media comprises a solid-state memory device.”

125. Olster discloses claim 14: “[t]he apparatus of claim 9, wherein the storage media comprises a solid-state memory device.”

126. Olster explains that the storage media 160 includes “solid-state drives (SSDs)” and “silicon storage devices,” which are types of *solid-state memory devices*. Ex. 1005, [0022]; *see also* my discussion above in Section VIII.A.2.

3. Claim 15: “The apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

127. Olster discloses claim 15: “[t]he apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.” *See* Section VIII.A.1.

IX. OPINIONS ON GROUND 2: CLAIMS 1-2, 7, 9, 14, AND 15 ARE OBVIOUS IN VIEW OF OLSTER IN COMBINATION WITH DAVIS

128. In my opinion, Olster in view of Davis discloses claims 1-2, 7, 9, 14, and 15. As I explained above, Olster describes a searchable data storage system that includes storage nodes that are made up of storage media and a controller, comprising a processor and memory. *See* Section VIII. Davis describes a policy for writing data to flash memory, based on the content being stored and the storage media. *See* my further discussion of Davis above in Section VII.B.

129. As explained below, a person of ordinary skill in the art would have combined Olster and Davis to take advantage of Davis's data placement policies. The data storage system resulting from the combination of the teachings of Olster and Davis would have included a modified version of the replication and placement component in Olster's storage policy that incorporate the data placement policies taught by Davis. Specifically, the replication and placement component would include policies that write data to a type of flash memory based on the data type. Ex. 1005, [0031] ("The replication and placement component of a policy defines the replication and placement of data objects in the data storage system."); Ex. 1008, 1:48-51 ("In some embodiments, a method for managing flash memory is provided. The method includes determining at least one property of a data and determining to which type of a plurality of types of flash memory to write the data, based on the at

least one property of the data.”), 9:37-11:15 (describing various data placement policies), FIG. 4A.

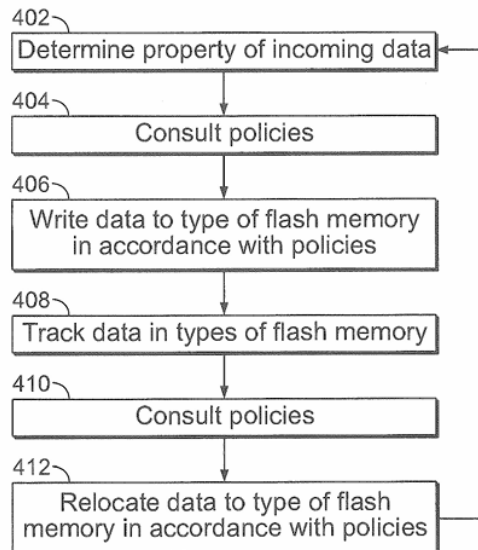


FIG. 4A

Ex. 1008, Fig. 4A

A. Motivation to Combine

130. In my opinion, a person of ordinary skill in the art would have been motivated to modify the replication and placement component of the storage policy of Olster (*at least one storage device policy*) to include the data placement policies taught by Davis, which records data to a type of flash memory based on the data type (*at least a recording method for writing content to the storage media*).

131. Both Olster and Davis are directed to data storage systems using solid-state memory that implement storage policies. Olster describes data “stored among storage nodes...according to [a] storage policy” that has a “replication and

placement component...[that] defines the replication and placement of data objects in the data storage system.” Ex. 1005, [0030], [0031]. Similarly, Davis describes storage device systems that “determine placement of data into types of flash memory, in accordance with properties of the data and various policies.” Ex. 1008, 3:32-34. In my opinion, a person of ordinary skill in the art implementing the data storage system of Olster would naturally look at other data storage systems that apply storage policies such as Davis to improve upon that system.

132. A person of ordinary skill in the art implementing the replication and placement component of Olster’s policy would have looked at other more detailed placement policies, like those described in Davis. Olster explains that “the replication and placement component may include specific distribution instructions for the data item...[which] may specify the number of zones in which the data item should be stored, the number of nodes in which the data items should be stored, a specific redundancy scheme identifier, and the like.” Ex. 1005, [0046]. But Olster provides no additional details that would guide such a person of ordinary skill in how to actually implement that policy. As such, a person of ordinary skill in the art would look at other systems that do include detailed replication and placement components, like Davis, in order to implement Olster’s policies.

133. A person of ordinary skill in the art would have been motivated to implement the data placement policies of Davis to leverage and optimize the

different types of storage media that are taught by Olster. Specifically, Davis teaches that “[t]rade-offs in cost per bit, bit density per unit of area in an integrated circuit, error rates, long-term storage reliability, read disturb rates, read wear rates and write wear rates, among other considerations, affect the choice of which type of flash memory to use in a given application.” Ex. 1008, 1:33-38. This aligns with what a person of ordinary skill in the art would have known, which is that different contexts require different configurations to be successful. Increasing the variety of storage media that are compatible with the system would be a benefit to the potential client or administrator. The data placement policies of Davis provide one known approach to place data in an optimized way to increase reliability and prioritize data.

B. Reasonable Expectation of Success

134. In my opinion, a person of ordinary skill in the art would have been able to modify the replication and placement component of Olster’s policy to include the data placement policies of Davis with a reasonable expectation of success.

135. The data storage systems of Olster and Davis have similar structures and components. Ex. 1005, [0030] (“In one embodiment, each object includes a storage policy identifier and a data portion. The object including its constituent data portion may be stored among storage nodes and storage zones according to the storage policy specified by the storage policy identifier included in the object.”); Ex. 1008, 3:30-34 (“Embodiments of a flash storage, a flash manager, and a related

method for managing flash memory, determine placement of data into types of flash memory, in accordance with properties of the data and various policies.”). Both Olster and Davis describe a controller that can be directed using a software application to execute memory access commands based on a storage policy. Ex. 1005, [0020] (“The storage zones 110, 120 and 104 may include a computing device and/or a controller on which software may execute. The computing device and/or controller may include...memories”); Ex. 1008, 20:12-20 (“It should be appreciated that, where a software-based embodiment is disclosed herein, the software can be embodied in a physical machine such as a controller. For example, a controller could...be configured to perform various actions, e.g., of a method, an application, a layer or an agent.”). Further, both store and retrieve data from storage media that includes flash memory. Ex. 1005, [0029] (“Storage devices include...flash memory devices”); Ex. 1008, Abstract, 2:4-13 (“In some embodiments, a flash storage device is provided....The device includes a flash manager, having at least one processor configured to perform actions that include writing a first data to the flash memory...”), FIG. 4B, 2:44-46 (describing “a flow diagram of a method for managing flash memory, applying a read rate threshold and a read count threshold”).

136. Both Olster and Davis utilize storage policies directed towards data placement. Ex. 1005, [0046] (“The replication and placement component may include specific distribution instructions for the data item”); Ex. 1008, 1:33-38

(Trade-offs in cost per bit, bit density per unit of area in an integrated circuit, error rates, long-term storage reliability, read disturb rates, read wear rates and write wear rates, among other considerations, affect the choice of which type of flash memory to use in a given application.”).

137. The data placement policies of Davis expand upon the replication and placement instructions of Olster, providing necessary details that Olster leaves up to a person of ordinary skill in the art’s discretion. Ex. 1005, [0031] (Example replication and placement policies include, full distribution, single copy, single copy to a specific zone, copy to all zone but a specified zone, and others.”), [0046]; Ex. 1008, 1:48-51 (“In some embodiments, a method for managing flash memory is provided. The method includes determining at least one property of a data and determining to which type of a plurality of types of flash memory to write the data, based on the at least one property of the data.”), 9:37-11:15 (describing multiple policy types).

138. Given the similar structures and components in Olster and Davis, the modification would be a simple programming change, which would be within the abilities of a person of ordinary skill in the art. In fact, Olster teaches that these policies can be “distributed from a client 102,” “system defined or early set by a system administrator,” effectively explaining to a person of ordinary skill in the art how to modify the policies of Olster. Ex. 1005, [0030], [0046].

C. Claims 1, 7, 9, and 14-15

139. As I discussed in Ground 1, Olster discloses each and every limitation of claims 1, 7, 9, and 14-15. *See* Section VIII.

140. To the extent that Olster does not expressly describe a *recording method for writing content to the storage media*, Olster in view of Davis renders obvious this limitation. Specifically, a person of ordinary skill in the art would have been motivated to modify the replication and placement component of Olster's policy to include the data placement policies taught by Davis. *See* Sections IX.A, IX.B.

141. The specification of the '023 patent only uses the term "recording method" once:

In an implementation, instead of varying the density of recording paths, the frequency of recorded bits along the paths may be varied. Similar to the concepts discussed above, if ***the frequency of the recording*** (i.e. bits per physical length, such as bits per inch) may be varied. By increasing the number of bits per inch that are recorded, storage capacity is increased, but data integrity may be degraded. In a further implementation, both of these variables may be adjusted to increase or decrease recording density. Regardless of the ***recording method chosen***, the device controller 130 can utilize the storage device policy to make these trade-offs.

Ex. 1001, 5:64-6:8. Therefore, the '023 patent explains that the *recording method* includes storing data based on at least recording density and priority level. This is confirmed by the dependent claims of the '023 patent that state that the recording

method can comprise “one or more of a recording path density or a recording frequency” (claim 5) or “instructions to determine a priority level for the content wherein the priority level indicates how densely the content is stored” (claim 6). On claim 5, while recording path density and recording frequency relate to hard disk drives, their recitation confirms that a *recording method for writing content to storage media* encompasses recording density.

142. Davis describes a *recording method for writing content to the storage media* because it describes the recording method embodiments described in the '023 patent. Specifically, both comprise a recording density and instructions to determine a priority level for the content indicating how densely the content is stored. Ex. 1008, 3:1-4 (“Controlling data placement into each of the two or more types of flash memory, based on data properties, supports selection of a type of flash memory for the data placement depending on how the data is used in the system.”).

143. As to recording density, Davis describes writing data based on types of flash memory, which “includes at least two types of flash memory having differing number of bits per cell.” Ex. 1008, 1:48-53. Davis explains that “[n]ow, flash memory is available in single level cell (SLC) and multilevel cell (MLC) types” where “[t]he single level cell type of flash memory has one bit per cell...and the multilevel cell flash memory types have more than one bit per cell.” *Id.*, 1:12-17. Therefore, Davis describes two different types of flash memory. “[T]he single level

cell type of flash memory has the lowest density, the highest cost per bit, the longest data retention time, the fastest average write time, the lowest read error rate, the lowest susceptibility to read disturb and program disturb (write disturb) events and errors, the lowest read wear rate and the lowest write wear rate.” *Id.*, 3:8-13. In contrast, “[h]igher numbers of bits per cell reverse these trends.” *Id.*, 3:13-14. As such, Davis explains that the data is written based on the different flash memory.

144. Davis describes storage policies that indicate a method for writing data to a specific type of flash memory based on its recording frequency (*a recording method for writing content to the storage media*). Ex. 1008, 9:31-36 (“data placement policies 326 applying to types of flash memory with differing numbers of levels per cell may be interpreted as or have corresponding data placement policies applying to types of flash memory with differing characteristics relative to data, and vice versa”).

145. As to priority level, Davis describes “[o]ne policy is writing text data to a type of flash memory having a lower read error rate and writing image data to a type of flash memory having a higher read error rate.” Ex. 1008, 9:37-39. The text data, which is more sensitive to read errors than image data, is given priority. *Id.* The text data with higher priority is stored less densely in the memory with lower bits per cell. Ex. 1008, 9:39-43 (“This policy could be interpreted as, or replaced by a policy of writing text data to a type of flash memory having a lower number of bits

per cell and writing image data to a type of flash memory having a higher number of bits per cell.”).

146. “This policy could be interpreted as...a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell.” Ex. 1008, 9:39-43. Thus, Davis describes storage policies that indicate a method for writing data that determines a priority level that indicates how densely the data is stored.

D. Dependent Claim 2

- 1. Claim 2: “The apparatus of claim 1, wherein the at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types and to assign a customized recording method to each of the one or more content types.”**

147. In my opinion, Olster in view of Davis renders obvious claim 2: “[t]he apparatus of claim 1, wherein the at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types and to assign a customized recording method to each of the one or more content types.”

148. As I discussed for 1[b], Olster describes a storage policy that includes a replication and placement component (*at least one storage device policy*). See Section VIII.A.3. In the combination, the replication and placement component further incorporates the data placement policies of Davis. As I explained, Davis

would include the detail that a person of ordinary skill in the art would want to effectively implement Olster's policies. *See* Section IX.A.

149. Davis describes *at least one storage device policy further comprises instructions that direct the device controller to differentiate the content by one or more content types* because its data placement policies are implemented by a controller executing instruction to identify data properties. Ex. 1008, Abstract (“The method includes determining at least one property of a data and determining to which type of a plurality of types of flash memory to write the data, based on the at least one property of the data.”), 20:17-20 (“a controller could include a first module and a second module and the controller could be configured to perform various actions, e.g., of a method, an application, a layer or an agent”). For example, Davis teaches differentiating between text data and image data:

One policy is writing text data to a type of flash memory having a lower read error rate and writing image data to a type of flash memory having a higher read error rate. This policy could be interpreted as, or replaced by a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell. This policy could also be written as two policies, one for text data and one for image data.

Ex. 1008, 9:37-45.

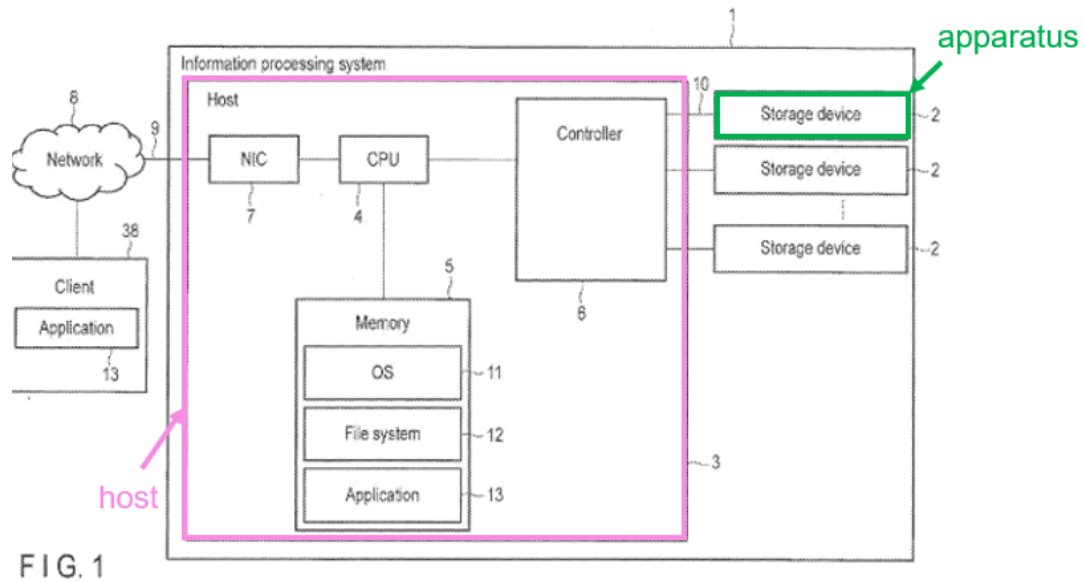
150. Davis further describes *at least one storage device policy further comprises instructions...to assign a customized recording method to each of the one or more content types* because Davis describes writing data to a type of memory

based on data properties. For example, in one policy, the controller assigns a method of writing to lower density type memory to text data and to higher density type memory to image data. Ex. 1008, 9:39-43. Therefore, the data properties (the content type) determine the method of writing, such as the density type of the memory.

X. GROUND 3: CLAIMS 1, 9, AND 14 ARE OBVIOUS IN VIEW OF HASHIMOTO

151. In my opinion, Hashimoto discloses claims 1, 9, and 14. Hashimoto describes a “storage device [that] includes a nonvolatile semiconductor memory device including a plurality of physical blocks, and a memory controller.” Ex. 1006, [0081]. The memory controller associates one or more of these physical blocks with stream IDs and executes write commands. *Id.*

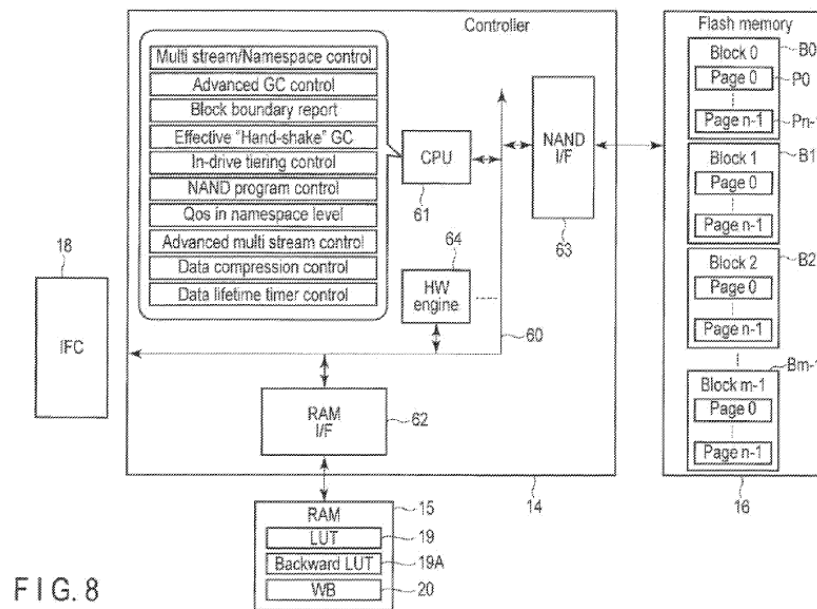
152. Hashimoto’s Figure 1 illustrates “an information processing system 1 including a storage device 2,” where the system “functions as a storage system that writes data to or reads data from the storage device 2.” *Id.*, [0082].



Ex. 1006, Fig. 1 (annotations in color)

153. This figure depicts that the information processing system includes a host 3 and one or more storage devices 2. *Id.*, [0083]. “[T]he storage devices 2 include solid state drives (SSD) 2 based on NAND flash technology.” *Id.*, [0084]. Storage device 2 includes a controller 14, RAM 15, flash memory 16, and an interface controller (IFC) 18, with the IFC 18 configured to transmit and receive signals to and from host 3 via interface 10. *Id.*, [0108]-[0109].

154. The controller 14 includes a CPU 61, RAM interface 62, NAND interface 63, and hardware engine 64, and functions as a flash translation layer (FTL) that manages the mapping between logical addresses and physical addresses of flash memory 16. *Id.*, [0148]-[0152].



Ex. 1006, Fig. 8

155. Hashimoto explains that host 3 “can manage and control the storage devices 2 by transmitting various commands to the storage device 2.” *Id.*, [0096]. Among the commands transmitted, Hashimoto describes tiering control that allows the host to designate a writing method associated with a recording density (SLC, MLC, TLC, or QLC). *Id.*, [0189]-[0195].

156. Hashimoto refers to the tiering control function as in-drive meaning that it is within the drive. As Hashimoto explains “[i]n-drive tiering control is a function of enabling a tier (first tier corresponding to a storage device having a feature of writing or reading data at a high speed and a tier (second tier) corresponding to another storage device having a feature of storing a large amount of data at a low cost, to exist together in the same SSD.” *Id.*, [0190].

157. Hashimoto discloses each limitation of claims 1, 7, 9, 14, and 15 as features in a single embodiment. I have been asked to assume that no motivation to combine or reasonable expectation of success is required when all features are disclosed in one embodiment as described in my analysis in Hashimoto. Nevertheless, it is my opinion that a person of ordinary skill in the art would have been motivated to combine the disclosed features in Hashimoto in a single system with a reasonable expectation of success.

158. A person of ordinary skill in the art would have been motivated to implement Hashimoto's data storage system with in-drive tier control because it would have maximized performance and endurance and minimized cost. Hot data requiring higher read and write speed can be stored using first tier writing methods whereas large amounts of cold data can be stored at a lower cost using second tier writing methods. Ex. 1006, [0199]. Similarly, data that may be more sensitive to errors (like text data) can be stored using more reliable first tier writing methods whereas less sensitive data (like image data) can be stored using less reliable second tier writing methods. *Id.*, [0201]. This customization provides optimization of the entire system.

159. A person of ordinary skill in the art would have had a reasonable expectation of success as Hashimoto itself teaches that in-drive tiering control is an option for its data storage system. Ex. 1006, [0451]-[0477], Figs. 40-41. Hashimoto

further teaches different ways the host-supplied settings could be implemented to include the in-drive tiering control. *Id.*, [0478]-[0506]. Such a system would have involved a combination of known technologies (data storage systems) according to known methods (communications) to yield the predictable result of a data storage system that implements host-supplied settings.

A. Independent Claims 1, 9

1. Preambles 1[pre]/9[pre]: “An apparatus comprising”

160. I have been asked to assume the preambles of claims 1 and 9 are limiting. Under that assumption, it is my opinion that Hashimoto discloses the preambles of claims 1 and 9, which recite “[a]n apparatus.”

161. Hashimoto describes a storage drive 2 (*apparatus*), such as a solid state drive (SSD). Ex. 1006, [0106] (“FIG. 4 shows a configuration example of the storage device 2.”), [0107] (“In the present embodiment, the storage device 2 is the SSD including the NAND flash memory (hereinafter called flash memory).”), Fig. 4; *see also* my discussion above in Section VII.C.

162. Hashimoto explains that “[t]he information processing system 1 includes a host (host device) 3, one or more storage devices 2, and an interface 10 configured to connect between the host (host device) 3 and the storage devices 2.” Ex. 1006, [0083]. Hashimoto further states that “[t]he storage devices 2 are semiconductor storage devices configured to write data to and read data from a

nonvolatile memory thereof. In the present embodiment, the storage devices 2 include solid state drives (SSD) 2 based on NAND flash technology.” *Id.*, [0084].

163. As I discuss below, storage device 2 (*apparatus*) comprises all the elements recited in the claims 1 and 9. *See* Sections X.A.2-11.

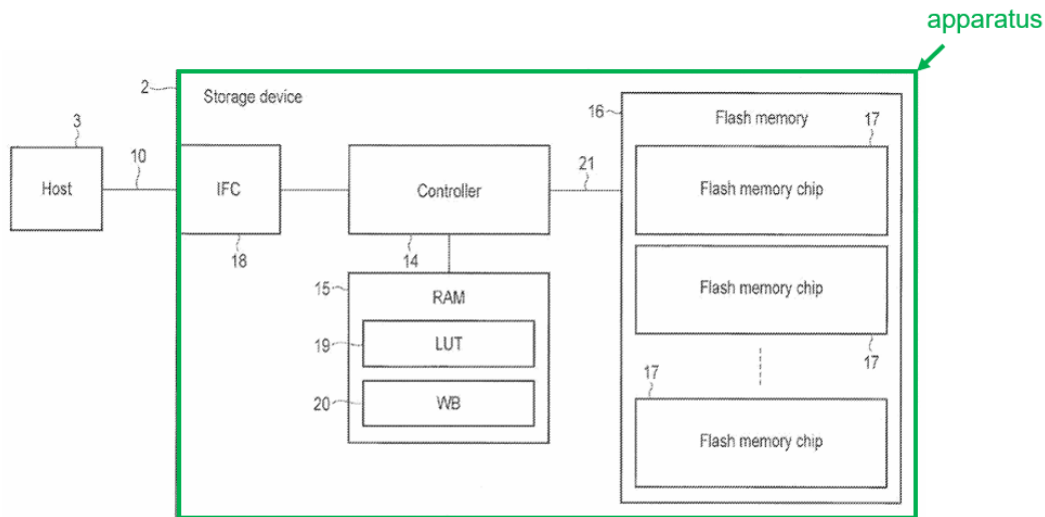


FIG. 4

Ex. 1006, Fig. 4 (annotations in color)

2. Limitations 1[a]/9[a]: “a storage media;”

164. In my opinion, Hashimoto discloses 1[a]/9[a]: “a storage media.”

165. Hashimoto describes storage device 2 (*apparatus*) that includes flash memory 16 (*storage media*) to store data. Ex. 1006, [0109] (“The flash memory 16 functions as a nonvolatile memory.”), [0195] (“The controller 14 of the storage device 2 writes the data to the flash memory 16 by employing the writing method designated by the tier attribute.”), Fig. 4.

166. Hashimoto further states that “the storage device 2 is the SSD including the NAND flash memory (hereinafter called flash memory).” *Id.*, [0107]. Flash memory would have been understood as a very commonly used type of *storage media* for data storage, as a discussed in my Overview of the Technology. See Section IV.A.

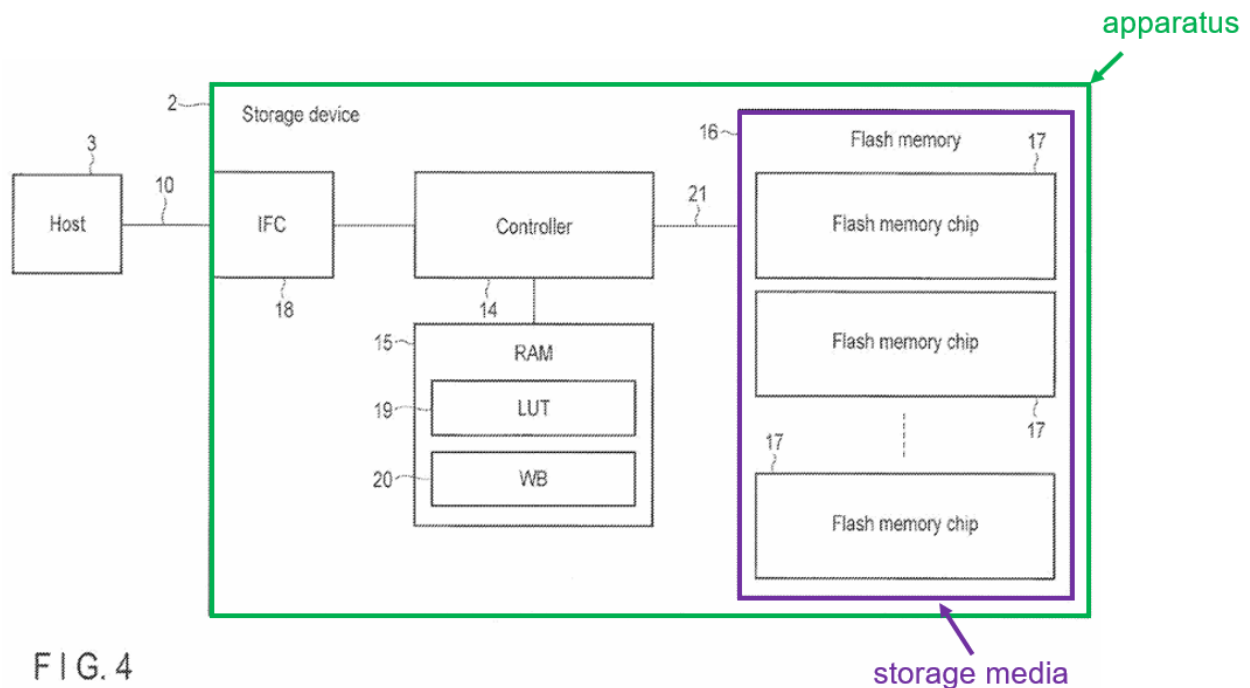


FIG. 4

Ex. 1006, Fig. 4 (annotations in color)

3. **Limitations 1[b]/9[b]: “a device controller operatively coupled with the storage media configured to at least:”**

167. In my opinion, Hashimoto discloses 1[b]/9[b]: “a device controller operatively coupled with the storage media” that is configured. See my further discussion below.

168. Hashimoto describes that storage device 2 (*apparatus*) includes controller 14 (*device controller*). Ex. 1006, [0035] (“FIG. 31 illustrates data structure of a get pending input block pool command sent to the storage device and return data of the get pending input block pool command and a processing sequence of the get pending input block pool command.”), Fig. 4. The controller 14 is *operatively coupled* with the flash memory 16 (*storage media*) because “[t]he controller 14 is electrically connected to the flash memory 16 via a flash memory interface 21 such as Toggle and ONFI.” *Id.*, [0114]. A person of ordinary skill in the art would have understood that two components being electrically connected via a flash memory would make them operatively coupled, because data could pass between the two components through the electrical connection. This is further supported by the inclusion of Toggle and ONFI, which are hardware interface standards used to transfer data between NAND flash memory and a controller.

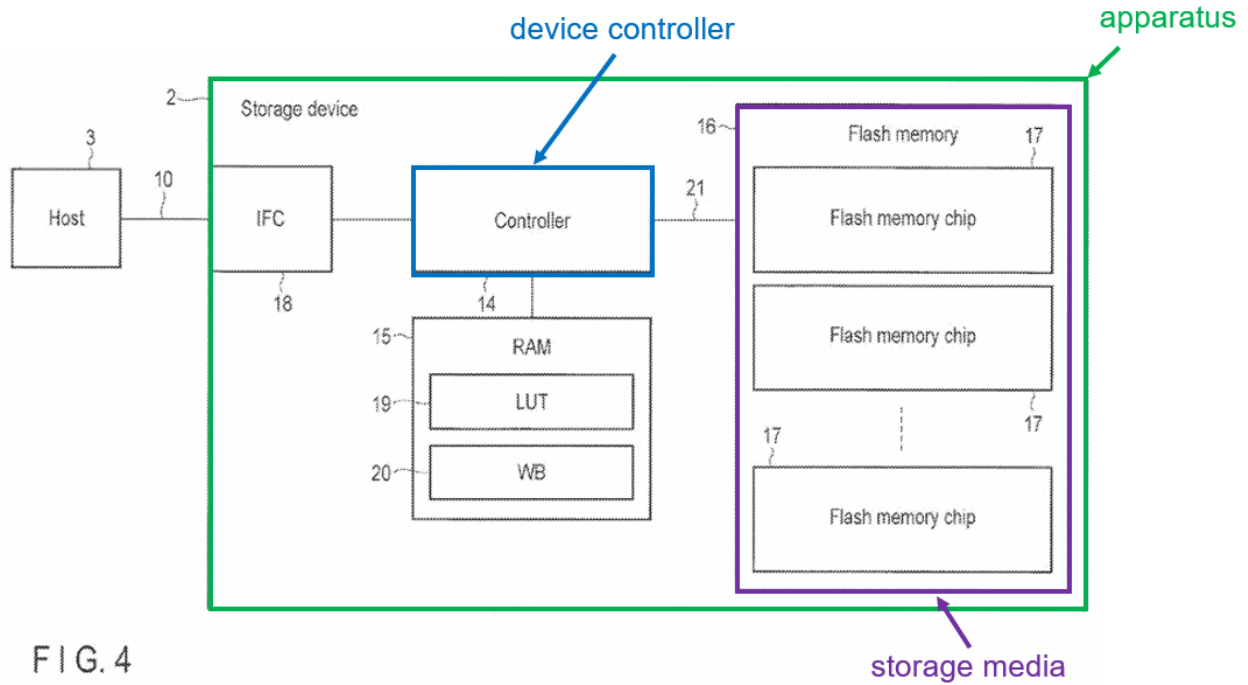
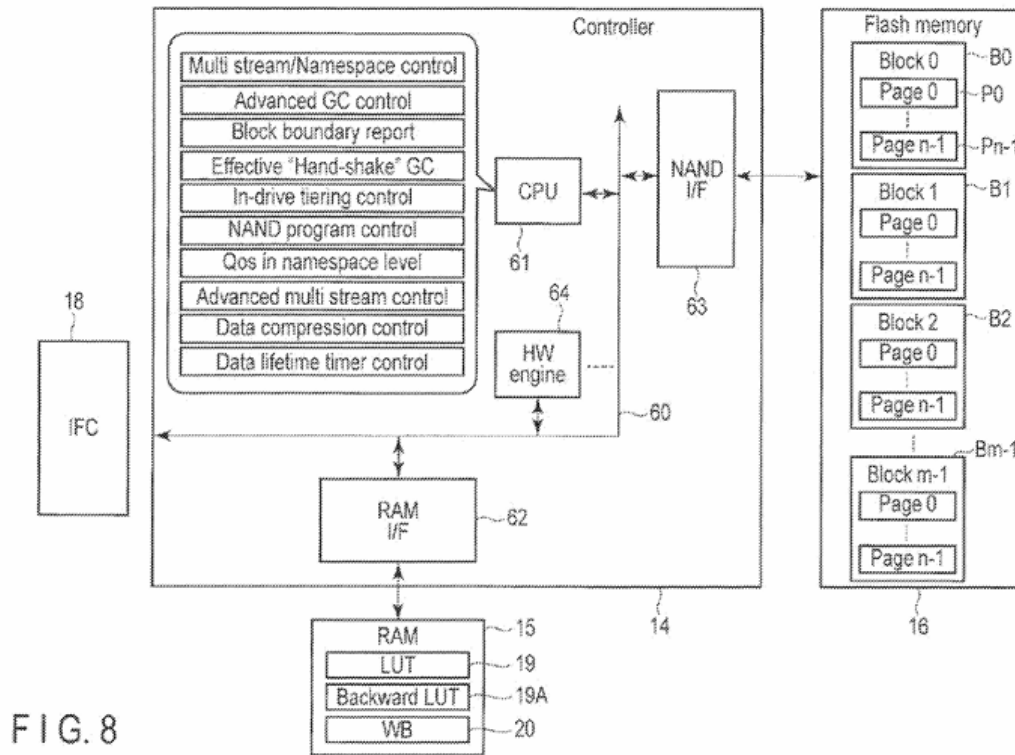


FIG. 4

Ex. 1006, Fig. 4 (annotations in color)

169. Hashimoto's Figure 8 provides further detail regarding the internal architecture of controller 14 and its operative connection to flash memory 16:



Ex. 1006, Fig. 8

170. With reference to this figure, Hashimoto describes that “[t]he controller 14 includes a CPU 61, a RAM interface 62, a NAND interface 63, and at least one hardware engine 64.” Ex. 1006, [0150]. Hashimoto describes that “[t]he CPU 61 is a processor configured to control the RAM interface 62, the NAND interface 63, and the hardware engine 64.” *Id.*, [0151]. Hashimoto further describes that “[t]he NAND interface 63 is a NAND controller configured to control the flash memory [16] under control of the CPU 61.” *Id.*, [0154]. Thus, controller 14 communicates with and controls data storage to flash memory 16 and accordingly describes the claimed “device controller operatively coupled with the storage media.”

171. As I discuss below, the controller 14 is *configured to at least* receive host-supplied settings (*receive, from a host, at least one storage device policy*) and store and retrieve the data (*content*) from the flash memory 16 (*storage media*) according to the host-supplied settings as I recite in 1[b.i]-1[b.ix] and 9[b.1]-9[b.viii].

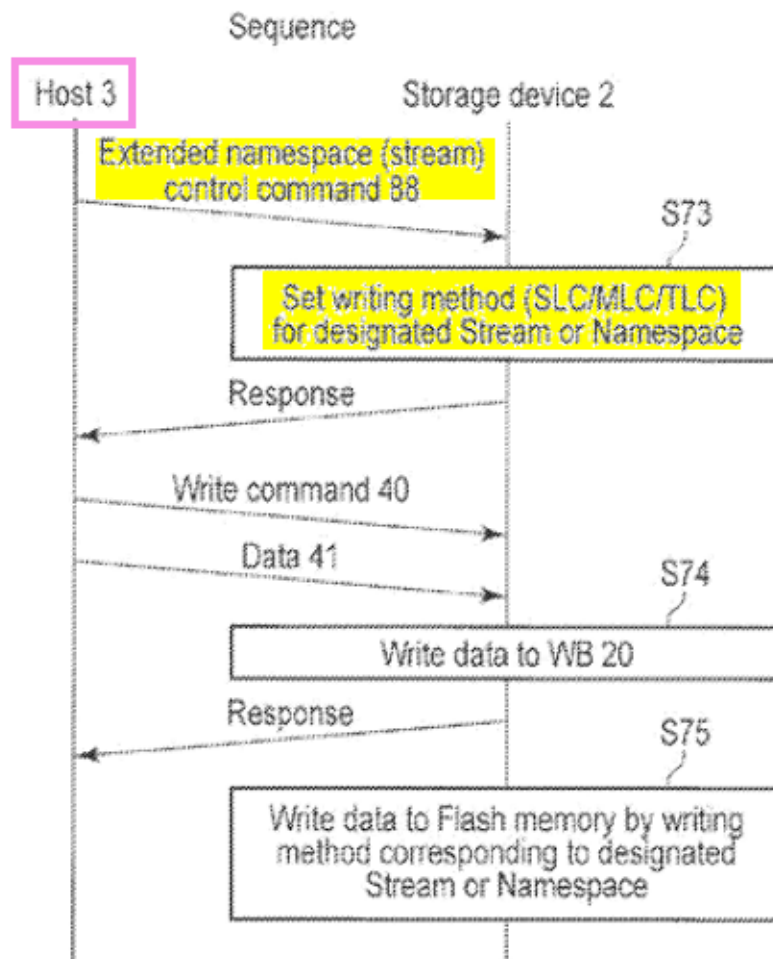
4. Limitations 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media;”

172. In my opinion, Hashimoto discloses 1[b.i]/9[b.i]: “receive, from a host, at least one storage device policy, wherein the at least one storage device policy indicates at least a recording method for writing content to the storage media.”

173. Hashimoto explains that controller 14 (*device controller*) receives commands from **host 3** (*from a host*). Specifically, Hashimoto explains that “[t]he host 3 (host software) can manage and control the storage devices 2 by transmitting various commands to the storage device 2.” Ex. 1006, [0096]. These commands (such as command 88) include to set writing and reading methods for certain data (*receive, from a host, at least one storage device policy*). Ex. 1006, [0491] (“The host 3 transmits the extended namespace (stream) control command 88 to the storage device 2 [and] [t]he controller 14 of the storage device 2 sets the writing method (SLC/MLC/TLC) for the designated target stream or target namespace (step

S73).”), Fig. 42 (“[s]et writing method (SLC/MLC/TLC) for designated Stream or Namespace”).

174. Hashimoto specifically describes that the host provides parameters that control how the SSD write content to the flash memory, such as identifying the writing method (SLC/MLC/QLC) for a stream or namespace that the controller later applies whenever writing data associated with the stream or namespace. *Id.*



Ex. 1006, Fig. 42 (excerpted and with annotations in color)

175. The commands in Hashimoto are host-supplied settings. Hashimoto further explains that these host-supplied settings can designate (*indicates*) a tier attribute to write the data to the flash memory 16 (*at least a recording method for writing data to the storage media*). Ex. 1006, [0193] (“The host 3 can designate a tier attribute which should be used to write the data, by using a command for In-drive tiering control (the extended write command or the extended namespace (stream) control command).”), [0195] (“The controller 14 of the storage device 2 writes the data to the flash memory 16 by employing the writing method designated by the tier attribute.”). This determines the location that the data is written into flash memory 16. A person of ordinary skill in the art would consider these host-supplied settings to be at least one storage policy, because they designate a recording method.

176. As I discussed in Ground 2, the specification of the ’023 patent only uses the term “recording method” once to describe recording methods that write data based on density in one embodiment and based on priority level in another embodiment. See Section IX. Hashimoto’s host-supplied settings (*at least one storage policy*) *indicates* the tier attribute (*at least a recording method for writing data to the storage media*) teaches both these ’023 patent embodiments of a *recording method for writing data to the storage media*. Ex. 1006, [0193], [0195].
Further:

The tier attribute applicable to the stream may be any one of the tier attribute for designating the SLC/MLC/TLC writing method, the tier attribute for designating the Faster write and shorter data retention/Slower write and longer data retention, and the tier attribute for designating the Faster write and slower read/Slower write and faster read...When the value of the Tier attribute is 00b, the controller 14 allocates the SLC block to the opened stream, and writes the data associated with the stream to the SLC block by the SLC writing method.

Id., [0614]-[0619].

177. **With respect to recording density**, Hashimoto explains that the tier attribute designates different write methods, where one tier uses less dense write methods (such as SLC write methods) and another tier uses denser write methods (such as MLC write methods. As Hashimoto further explains, “[i]n the first tier, a 1-bit/cell writing method (SLC writing method) of writing 1-bit data to each memory cell is employed.” Ex. 1006, [0191]. “In the second tier, a writing method of writing data of two or more bits to each memory cell is employed.” *Id.*; *see also id.* (“The writing method employed in the second tier may be a 2-bit/cell writing method (MLC writing method), a 3-bit/cell writing method (TLC writing method), or a 4-bit/cell writing method (QLC writing method).”). Therefore, because the tier attribute is associated with a writing method, the tier attribute determines which writing method (such as SLC, MLC, TLC, or QLC writing method) is used to write data to the flash memory 16.

178. Hashimoto describes an in-drive tiering feature that enables the host to direct the controller to store data at different densities in accordance with various “writing methods.” *Id.*, [0011], [0135]-[0137], [0189]-[0195]. Hashimoto explains that “[i]n-drive tiering control is a function of enabling a tier (first tier) corresponding to a storage device having a feature of writing or reading data at a high speed and a tier (second tier) corresponding to another storage device having a feature of storing a large amount of data at a low cost, to exist together in the same SSD.” *Id.*, [0189]-[0190].

179. Hashimoto states that “[i]n the first tier, a 1-bit/cell writing method (SLC writing method) of writing 1-bit data to each memory cell is employed,” and “[i]n the second tier, a writing method of writing data of two or more bits to each memory cell is employed. The writing method employed in the second tier may be a 2-bit/cell writing method (MLC writing method), a 3-bit/cell writing method (TLC writing method), or a 4-bit/cell writing method (QLC writing method).” *Id.*, [0191]-[0192]. Hashimoto’s writing methods thus enable the controller to store data at different densities (*e.g.*, with 1-4 bits per cell).

180. Hashimoto’s in-drive tiering policy is host-supplied: “The host 3 can designate a tier attribute which should be used to write the data, by using a command for In-drive tiering control (the extended write command or the extended namespace (stream) control command).” *Id.*, [0193]. With reference to Figure 42, Hashimoto

describes an extended namespace (stream) control command 88 that sets the density policy prior to write commands.

181. Hashimoto explains that the extended namespace (stream) control command 88 “is a control command capable of designating a tier attribute. More specifically, the extended namespace (stream) control command 88 designates the tier attribute to be applied in units of not write data, but stream or namespace.” *Id.*, [0479]. This command 88 “may be, for example, an open stream command or a create namespace command” (*id.*, [0480]) including the following input parameters: “(1) Stream ID (or Namespace ID); (2) Tier attribute,” where “[t]ier attribute indicates the writing method which should be employed for the target stream or the target namespace.” *Id.*, [0482]-[0485]. Based on the command 88 received from the host 3, the “controller 14 of the storage device 2 sets the writing method (SLC/MLC/TLC) for the designated target stream or target namespace (step S73).” Ex. 1006, [0491]. The controller’s act of “set[ting] the writing method” for the stream or namespace confirms that the host-supplied policy is received and stored on the device controller.

182. Hashimoto further confirms that this stored policy directs how the controller handles content-storage decisions for subsequent write operations. After the extended namespace (stream) control command 88 has been received and the writing method has been set, “[t]he host 3 transmits the write command 40 including

the namespace ID or the stream ID to the storage device 2 and also transmits the write data 41 to the storage device 2.” *Id.*, [0492]

183. **With respect to priority level**, Hashimoto explains that the tier attribute can also determine a priority level for the data to indicate how densely the data should be stored. Ex. 1006, [0198] (“The host 3 can designate a tier attribute giving a higher priority to the write speed than to data retention or a tier attribute giving a higher priority to the data retention than to the write speed, by using a command for the NAND program control (Extended write command, etc.).”).

184. For example, Hashimoto describes that tier attribute determines priority based on whether the data is hot or cold. Ex. 1006, [0199] (“[W]hen data (hot)...the host 3 may designate the tier attribute giving a higher priority to the write speed than to the data retention. In contrast, when data (cold)...the host 3 may designate the tier attribute giving a higher priority to the data retention than to the write speed.”). Hot data is being updated much more frequently than cold data, therefore, the write speed will be more important for hot data that is updated or retrieved more frequently. Alternatively, cold data that is not interacted with as frequently will have more of an emphasis on data retention to ensure that cold data is maintained over a longer period of time. As another example, “when data is required to have high reliability, the host 3 may designate a tier attribute giving a higher priority to the data reliability than to the read speed.” Ex. 1006, [0202].

5. Limitation 1[b.ii]: “receive a storage request, from the host, to store content;”

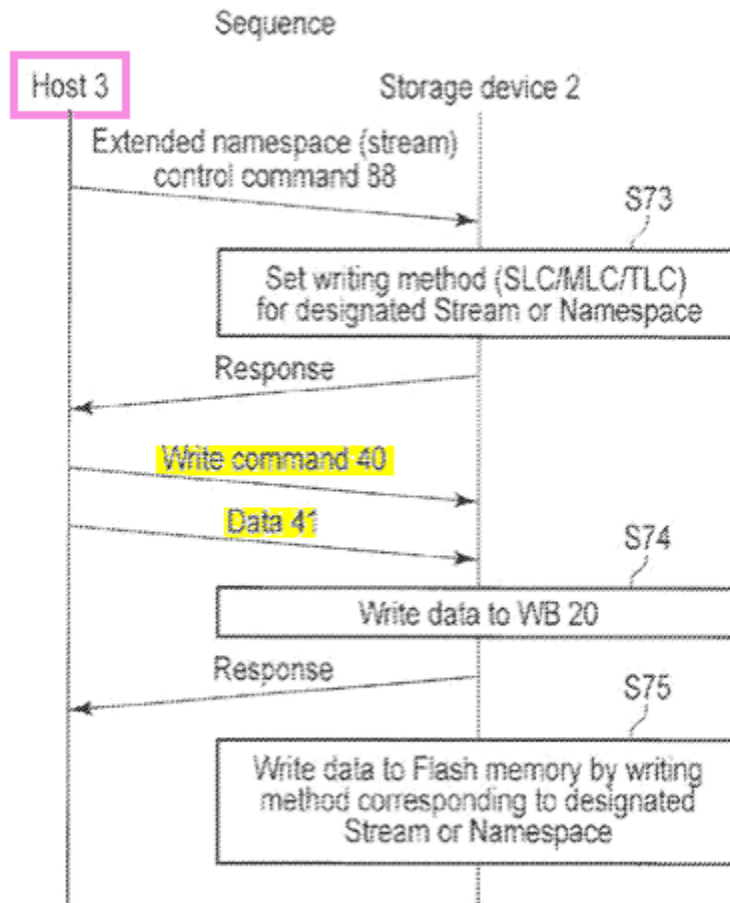
185. In my opinion, Hashimoto discloses 1[b.ii]: “receive a storage request, from the host, to store content.”

186. Hashimoto describes that controller 14 (*device controller*) receives write command 40 (*a storage request*), from **host 3** (*from the host*) to store data 41 (*to store content*). Ex. 1006, [0492] (“The host 3 transmits the write command 40...to the storage device 2 and also transmits the write data 41 to the storage device 2.”), Fig. 42 (illustrating the fact that the host 3 transmits a write command 40 containing data 41). The controller 14 is the part of storage device 2 that receives write command 40 because it executes the command. *Id.* (“[T]he controller 14 writes the write data to the block in the flash memory 16 by the writing method for the designated target stream or target namespace (step S75).”); *see* Section X.A.3. Hashimoto further describes the write command and its receipt by the controller 14:

The host 3 transmits a write command to the storage device 2. This command includes the logical address (starting logical address) and a sector counter of the data to be written. The LBAs are used as the logical addresses in the present embodiment, but an object ID may be used as the logical addresses in other embodiments.... The controller 14 of the storage device 2 writes the data to be written designated by the Starting LBA and the sector count in the write command, to the block in the flash memory 16. Furthermore, the controller 14 updates the lookup table (LUT) 19 to map the LBAs corresponding to the written data to the physical addresses corresponding to physical storage

locations at which the data is written.

Id., [0121]. Indeed, the controller receives the starting logical address as part of the write command and uses it to write data designated by that address. *Id.* Hashimoto accordingly discloses that its controller 14 (*device controller*) receives the write command (*storage request*) from the host.



Ex. 1006, Fig. 42 (excerpted and with annotations in color)

187. Hashimoto’s write command is a “storage request ... to store content.” Specifically, with reference to Figure 11, Hashimoto discloses that the host 3 transmits write command 40 and write data 41 (*content*) to the storage device 2.

Hashimoto states that the “write command 40 includes a parameter indicating a starting LBA of the write data [41] and a parameter indicating a sector count (transfer length) of the write data [41].” *Id.*, [0251].

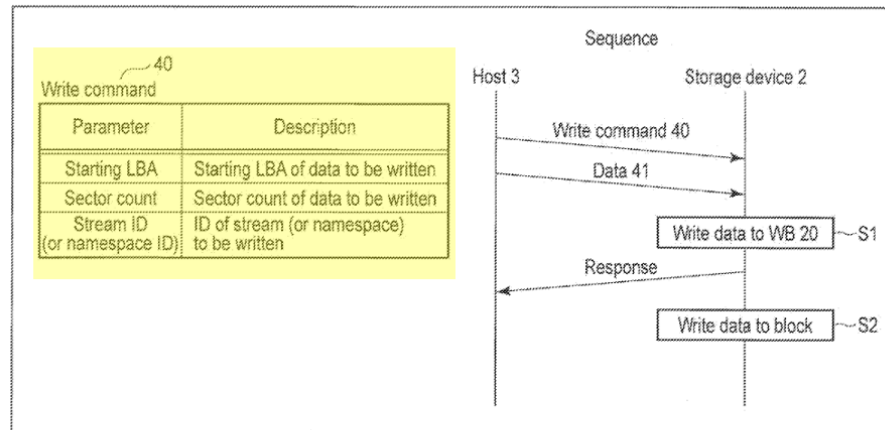
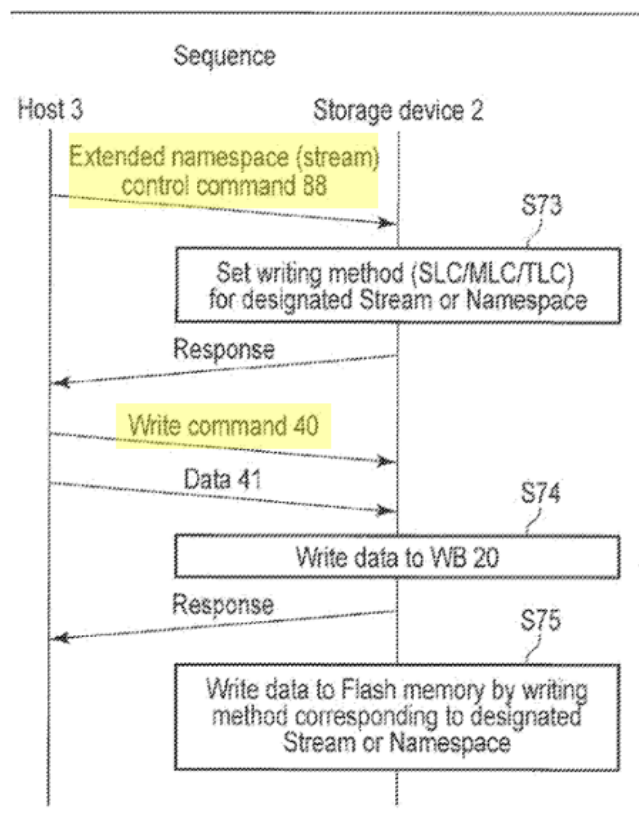


FIG. 11

Ex. 1006, Fig. 11 (excerpted and with annotations in color)

188. Hashimoto also discloses that the controller 14 receives from the host 3 a write command 40 (*a storage request*) and write data 41 (*content*) in the application of its density-control storage device policy. As shown in Figure 42, after the extended namespace (stream) control command 88 (*storage device policy*) has “[s]et [the] writing method (SLC/MLC/TLC) for designated Stream or Namespace,” the host 3 then “transmits the write command 40 including the namespace ID or the stream ID to the storage device 2 and also transmits the write data 41 to the storage device 2.” *Id.*, [0492]. Upon receiving this write command, “the controller 14 writes the write data to the write buffer (WB) 20 (step S74), and sends to the host 3 the

notice of command completion. After that, the controller 14 writes the write data to the block in the flash memory 16 by the writing method for the designated target stream or target namespace (step S75).” *Id.* Hashimoto’s host-issued write command 40, which directs controller 14 to store accompanying write data 41 in flash memory 16, thus discloses the claimed *storage request ... to store content*.



G. 42

Ex. 1006, Fig. 42 (excerpted and with annotations in color)

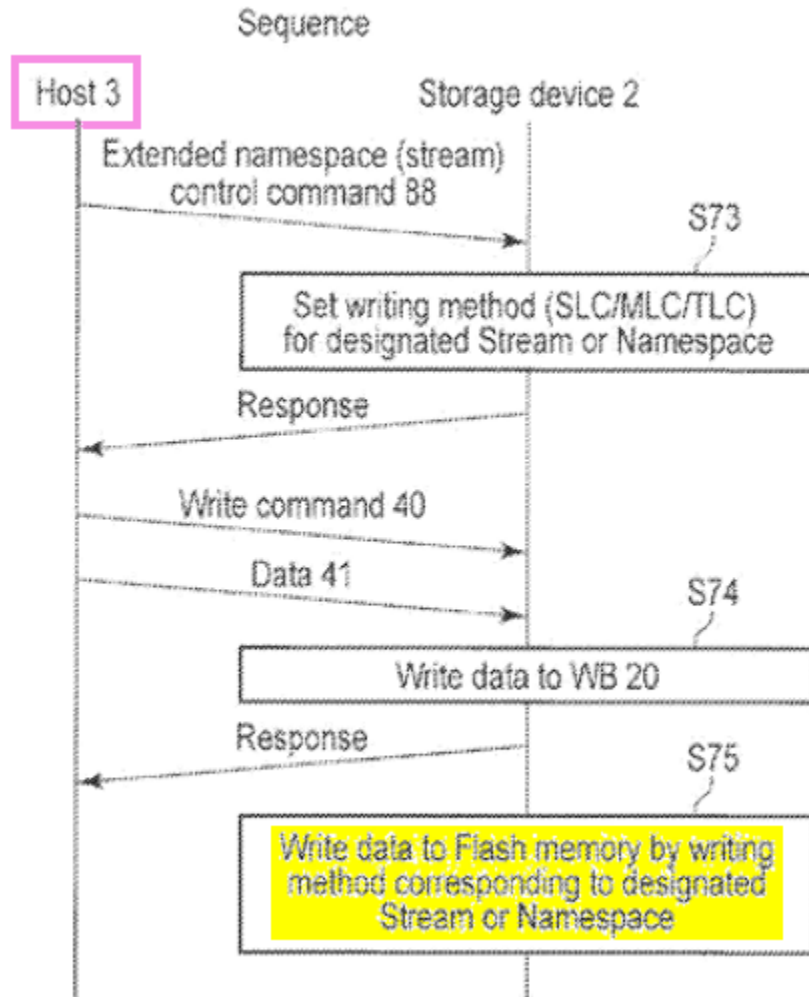
6. Limitations 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”

189. In my opinion, Hashimoto discloses 1[b.iii]/9[b.ii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy.”

190. Hashimoto explains that controller 14 (*device controller*) writes the data (*store the content*) on flash memory 16 (*on the storage media*). Hashimoto writes the data (*stores the content*) based on the writing method indicated by the tier attribute (*by utilizing the recording method*) designated in the host-supplied settings (*indicated by the at least one storage device policy*). Specifically, Hashimoto describes that “controller 14 writes the write data to the block in the flash memory 16 by the writing method for the designated target stream or target namespace (step S75).” Ex. 1006, [0492], Fig. 42. In other words, the controller stores the data in flash memory by applying the writing method that was previously set by the tier attribute settings included in the extended namespace (stream) control command 88.

191. As I discussed for 1[b.i]/9[b.i], the writing method for the designated target stream is determined by the tier attribute set in step S73. *See* Section X.A.4; Ex. 1006, [0492], Fig. 42. Therefore, because the tier attribute determines the

writing method, the tier attribute set in step S73 determines how the data (*content*) is written (*stored*) on flash memory 16 (*on the storage media*).

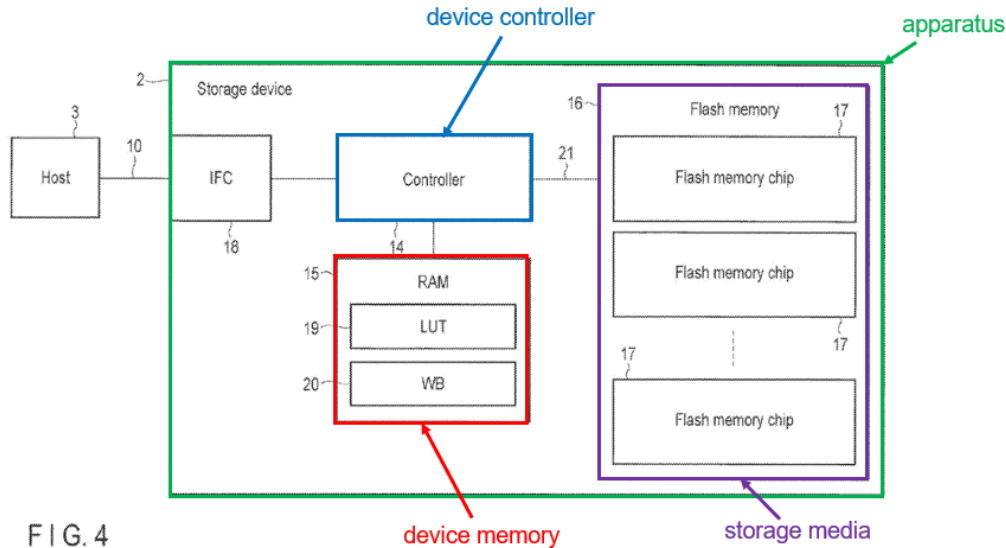


Ex. 1006, Fig. 42 (excerpted and annotations in color)

7. Limitations 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”

192. In my opinion, Hashimoto discloses 1[b.iv]/9[b.iii]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content.”

193. Hashimoto explains that the “RAM 15” is a *memory of* “controller 14” (*device controller*). For example, CPU 61 of controller 14 exclusively controls **RAM 15**. Ex. 1006, [0150] (“[t]he controller 14 includes a CPU 61, a RAM interface 62”), [0152] (“The RAM interface 62 is a RAM controller configured to control the RAM 15 under control of the CPU 61.”), Figs. 4, 8. “The RAM 15 includes a storage region to store a lookup table (LUT) 19.” *Id.*, [0109]. A person of ordinary skill in the art would have known that controller 14’s exclusive control over RAM 15 would render RAM 15 the controller 14’s memory (*device controller’s memory*).



Ex. 1006, Fig. 4 (annotations in color)

194. Hashimoto also describes that controller 14 (*device controller*) stores (*record*) mapping and processing information associated with data 41 (*storage information for the content*) in LUT 19 in RAM 15 of the device controller (*a memory of the device controller*). Hashimoto teaches that this mapping and processing information indicates a relationship between logical addresses and physical addresses, to understand the locations in flash memory 16. *Id.*, [0110] (“mapping information indicating relationship between logical addresses and physical addresses of the flash memory 16.”). This logical-to-physical (L2P) address mapping (part of the *storage information*) is *for* write data 41 (*content*) because the controller updates the LUT 19 to map logical addresses (LBAs) associated with the data to physical addresses after storing that data in flash memory 16, including in the application of the in-drive tiering. *Id.*, [0121] (“Furthermore,

the controller 14 updates the lookup table (LUT) 19 to map the LBAs corresponding to the written data to the physical addresses corresponding to physical storage locations at which the data is written.”), [0254] (“[T]he controller 14 writes the write data 41 to...the flash memory 16 (step S2). The controller 14 updates the lookup table 19 and maps the LBA of the write data to the physical address of the physical storage location at which the write data has been written.”), [0492].

195. Hashimoto also explains that processing information about how data 41 is stored (part of *storage information for the content*) is included in LUT 19. Ex. 1006, Fig. 4 (as annotated above). For example, as part of the tier attribute, “host 3 can designate either a tier attribute of compressing data and writing the compressed data to the block of the flash memory 16 or a tier attribute of writing the write data to the block without compressing the data.” Ex. 1006, [0222]. Therefore, the host 3 can designate that the processing information is included in LUT 19. The associated processing information, such as compression, is also stored in LUT 19. Ex. 1006, [0662] (“The controller 14 updates the lookup table 19, maps the LBA of the write data to the physical address of the physical storage location at which the write data has been written, and further associates a compression flag to the LBA of the write data”), [0663]-[0665], Fig. 61.

LUT 19

Logical addresses (LBA)	Physical addresses	Compression flags	Others

FIG. 61

Ex. 1006, Fig. 61 (annotations in color)

196. Hashimoto explains that the L2P address mapping stored on RAM 15 *compris[es] at least the LBA (a content identifier) for the data (for the content)*. Hashimoto describes that the LUT 19 of RAM 15 records LBAs as part of its L2P address mapping. *Id.*, [0416] (“The controller 14 acquires information about the logical addresses corresponding to the valid data stored in the target block for the next garbage collection, by referring to the lookup table 19.”), [0434], [0662] (“lookup table 19, maps the LBA of the write data to the physical address of the physical storage location at which the write data has been written”).

197. An LBA identifies data because the LBA is unique identifier used by the host to subsequently retrieve that data from the flash memory based on that

unique, individualized relationship. *Id.*, [0104] (“[t]he command such as a write command and a read command includes a starting logical address (Starting LBA: Logical Block Address) corresponding to the data (data to be written or data to be read)”), [0121] (“[t]he controller 14 of the storage device 2 writes the data to be written designated by the Starting LBA and the sector count in the write command”), [0416] (“ the controller 14 provides the host 3 with a list (LBA list) of the logical addresses corresponding to the valid data stored in the target block for the next garbage collection”), [0417] (“[t]he host 3 can transmit a command (trim command or unmap command) designating a logical address range of the discardable data”), [0434] (“[t]he controller 14 acquires the list of all LBAs mapped to the target active block for the next garbage collection, i.e., the list of LBAs corresponding to the valid data in the target active block”), [0437] (“The host 3 creates a list of LBAs corresponding to the discardable data”); *see* Section VI.C. Therefore, the LBA is *a content identifier for the content*.

8. Limitations 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content;”

198. In my opinion, Hashimoto discloses 1[b.v]/9[b.iv]: “subsequent to storing the content, receive a content request for the content, comprising at least the content identifier for the content.”

199. Hashimoto explains that after the data 41 is written to flash memory 16 (*subsequent to storing the content*), the controller 14 (*device controller*) can then receive a read command (*content request*) for data 41 (*content*). Ex. 1006, [0663] (“controller 14 receives the read command from the host 3”); *see also id.* [0082] (“The information processing system 1 functions as a storage system that writes data to or reads data from the storage device 2.”), [0098] (“In the present embodiment, not only general commands such as a write command, a read command...are transmitted to the storage devices 2 via the interface 10.”), cl. 17 (“a host configured to send commands to write data to or read data from the storage device”).

200. As I discussed previously, Hashimoto explains that the LBA is a *content identifier*. See Section X.A.7. The read command for data 41 (*content request for the content*) includes the LBA (*content identifier*) for the data 41: “The command such as a write command and a read command includes a starting logical address (Starting LBA: Logical block Address) corresponding to the data (data to be written or data to be read) and a sector count (transfer length).” *Id.*, [0104]. Therefore, the read command includes the LBA as a way of identifying the data 41 to be read.

201. Hashimoto also explains that the read command is received *subsequent to storing the content* because the controller uses the same LBA for both reads and writes (*id.*, [0104]), which means that the controller can only satisfy the request by

consulting the L2P address mapping already established for that data at write time. For example, “[w]hen the controller 14 receives the read command from the host 3, the controller 14 determines whether or not the data requested to be read is compressed, by referring to the lookup table 19,” and if so, “de-compresses the data read from the flash memory 16.” *Id.*, [0663]. This determination is only possible because the L2P address mapping of the data requested by the read command (*content request for the content*) was previously recorded in the LUT 19 in the context of a write command (*storage request*).

9. Limitations 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content;”

202. In my opinion, Hashimoto discloses 1[b.vi]/9[b.v]: “in response to the content request, retrieve from the memory of the device controller, the storage information for the content.”

203. As I discussed previously, Hashimoto describes a read request (*content request*) (Section X.A.8) and L2P mapping (*storage information for the content*) that are stored in LUT 19 in RAM 15 of controller 14 (*memory of the device controller*) (Section X.A.7).

204. Hashimoto explains that *in response to the read command (content request)*, controller 14 *retrieves from* LUT 19 in RAM 15 of the controller 14 (*memory of the device controller*), the L2P mapping and processing information

(*storage information for the content*). As Hashimoto further explains, “[w]hen the controller 14 receives the read command” (*content request*) “from the host 3, the controller 14” (*device controller*) “determines whether or not the data requested to be read is compressed, by referring to the lookup table 19” of RAM 15 (*storage information for the content*). Ex. 1006, [0663].

205. The controller 14 then *retrieves the* L2P address mapping and processing information (*storage information*) *from* LUT 19 of RAM 15 (*the memory of the device controller*). The L2P mapping and processing information was previously stored in LUT 19 of RAM 15 and “referring to [the L2P mapping in] the lookup table 19” and other processing information necessitates retrieving it first. *Id.*, [0663]; *see* Section X.7. The controller 14 will need the L2P mapping to identify where the data is stored and the processing information (such as whether the data is compressed to determine whether the data needs to be decompressed and the ECC applied) to recover the data.

10. Limitations 1[b.viii]/9[b.vii]: “recover the content from the storage media, [at least in part according to the storage information]/[according to the storage information and according to the at least one storage device policy]; and”

206. In my opinion, Hashimoto discloses 1[b.viii]/9[b.vii]: “recover the content from the storage media, [at least in part according to the storage

information]/[according to the storage information and according to the at least one storage device policy].”

207. As I discussed above, Hashimoto describes that controller 14 (*device controller*) stores the data (*content*) on the flash memory 16 (*storage media*). See Section X.A.6. As I further discussed, Hashimoto describes retrieving the L2P mapping (*storage information*) in response to a read command (*content request*). See Section X.A.9.

208. Hashimoto explains that controller 14 (*device controller*) retrieves data 41 (*recover the content*) from the flash memory 16 (*storage media*) at least in part according to the L2P mapping and processing information, such as whether the data is compressed (*at least in part according to the storage information*). Controller 14 can only read the requested data “from the flash memory 16” because the physical address of that data was previously retrieved from the L2P address mapping (*storage information*) stored in LUT 19. *Id.*, [0110] (LUT 19 “stores mapping information indicating relationship between logical and physical addresses of the flash memory 16”), [0663] (“When the controller 14 receives the read command from the host 3, the controller 14 determines whether or not the data requested to be read is compressed, by referring to the lookup table 19.”).

209. After “referring to the lookup table 19[,] [i]f the data requested to be read is compressed, the [controller] 14 de-compresses the data read from the flash

memory 16 and transmits the de-compressed data to the host 3.” Ex. 1006, [0663].

Therefore, Hashimoto describes *recovering* the data (*content*) from flash memory 16 (*the storage media*), *at least in part according to* the L2P mapping and processing information (*storage information*).

210. Regarding claim 9 specifically, Hashimoto explains that controller 14 (*device controller*) also retrieves data 41 (*recover the content*) *from the* flash memory 16 (*storage media*) *according to the* host-supplied settings including the tier attribute (*at least one storage device policy*). *See* Section X.A.4. As I discussed for 1[b.1]/9[b.i], the tier attribute determines the writing method (SLC/MLC/TLC) used to write the data. *See* Section X.A.4. To retrieve the data, Hashimoto explains that controller 14 must also determine the writing method used to read the data from memory.

211. For example, “[w]hen a write method (SLC: Single Level Cell) for writing 1-bit data per memory cell is employed, the physical sector 35 is equivalent to a page.” Ex. 1006, [0134]-[0135]. Alternatively, “[w]hen a write method (MLC: Multi Level Cell) for writing 2-bit data per memory cell is applied, the physical sector 35 is equivalent to two pages (i.e., an upper page and a lower page).” *Id.* Hashimoto explains that “[o]ne word line and one physical sector are selected based on the physical address like the row address received from the controller 14, during the read operation.” *Id.*, [0135]. To determine whether the address includes one

page or two pages (based on whether the storage technology is SLC or MLC), controller 14 uses the host-supplied settings including the tier attribute (*at least one storage device policy*) to determine which writing method was used.

212. Hashimoto further describes that controller 14 (*device controller*) retrieves data 41 (*recover the content*) from the flash memory 16 (*storage media*) according to the host-supplied settings including the tier attribute (*at least one storage device policy*). The tier attribute is used to determine the error correction code (ECC) that controller 14 applies to detect errors in the data before returning the data to the host, at which point the data 41 (*content*) is retrieved (*recovered*). Hashimoto explains that the ECC is determined “in accordance with the designated tier attribute,” so the ECC is determined *according to at least one storage policy*. Ex. 1006, [0521]. For example, when controller 14 selects a “writing method giving a higher priority to the data retention than to the write speed” according to the tier attribute, a “a strong ECC is applied to assure the data reliability.” Ex. 1006, [0522]-[0523]. In response to a read command, controller 14 applies the ECC decoding based on host-supplied settings including the tier attribute, such that the *storage device policy* and the *storage information* are relied upon to *recover the content*.

11. Limitations 1[b.ix]/9[b.viii]: “respond to the content request at least by returning the content to the host.”

213. In my opinion, Hashimoto discloses 1[b.ix]/9[b.viii]: “respond to the content request at least by returning the content to the host.”

214. As I discussed previously, Hashimoto explains that in response to a read command (*respond to the content request*), the controller 14 (*device controller*) recovers the *content* from the flash memory 16. See Sections X.A.9-10. Hashimoto's controller 14 (*device controller*) responds to the read command (*respond to the content request*) by *at least returning the* ECC-decoded data (*content*) to host 3 (*the host*). Ex. 1006, [0663] (controller 14 transmits requested data to the host 3); *see also id.*, [0445] ("When the requested data is stored in the read cache area 133 (cache hit), the file system 12 transmits a read command to the storage device 2 (step S154) and receives the requested data from the storage device 2 (step S155)."), [0449] ("When the requested data is stored in lower tier area 144, the file system 12 transmits a read command to lower tier storage device 140 (step S186), receives the requested data from lower tier storage device 140 (step S187).").

B. Dependent Claim 14:

1. Claim 14: "The apparatus of claim 9, wherein the storage media comprises a solid-state memory device."

215. In my opinion, Hashimoto discloses claim 14: "[t]he apparatus of claim 9, wherein the storage media comprises a solid-state memory device."

216. As I discussed above, Hashimoto describes storage drive 2 (*apparatus*), such as a solid-state device (SSD), that includes flash memory 16 (*storage media*). See Section X.A.2. Hashimoto explains that the "flash memory 16 includes one or

more flash memory chips 17. Ex. 1006, [0109]; *see id.*, [0084] (“The storage devices 2 are semiconductor storage devices....[T]he storage devices 2 include solid state drives (SSD) 2 based on NAND flash technology.”), [0107] (“The storage device 2 is the SSD including the NAND flash memory (hereinafter called flash memory).”). As a person of ordinary skill in the art would have known, solid-state memory, also referred to as nonvolatile memory, is a type of memory (such as NAND flash memory) that stores data persistently without moving parts.

217. Therefore, flash memory 16 (*storage media*) comprises a *solid-state memory device* in the form of flash memory chip 17.

XI. GROUND 4: CLAIMS 7 AND 15-20 ARE OBVIOUS IN VIEW OF HASHIMOTO AND KILIAN

218. In my opinion, the combination of Hashimoto and Kilian discloses claims 1, 7, 9, and 15-20. Hashimoto describes a “storage device [that] includes a nonvolatile semiconductor memory device including a plurality of physical blocks, and a memory controller.” Ex. 1006, [0081]. Kilian describes a system and method for retention of data on a data storage system. Ex. 1007, Abstract. As I discuss below, a person of ordinary skill in the art would have combined the data retention methods described in Kilian with the data storage system described in Hashimoto.

A. Motivation to Combine

219. Kilian identifies benefits of its host-supplied data retention feature. Kilian explains that traditional data protections are inadequate because “an

administrator of the system typically has the ability to change the ‘read-only’ status of files such that the files may be modified or deleted,” and “conventional file systems do not allow the user to specify a period of time for which data cannot be deleted or modified.” Ex. 1007, 1:38-42. Kilian’s retention period addresses this issue by allowing greater configurability: “once the retention period has been established for a unit of data, the storage system ensures that the unit of data is retained in an unmodified state for the full retention period.” *Id.*, 3:36-39.

220. A person of ordinary skill in the art would have been motivated to add Kilian’s host-specified data retention/write-protection functionality to Hashimoto’s host-programmable controller to obtain these expressly-taught benefits—*i.e.*, enforceable, time-bounded protection against accidental or malicious deletion or modification at the storage-device level—with a reasonable expectation of success, as it requires only programming a known retention rule into a controller already designed to receive host policies.

221. A person of ordinary skill in the art would also have recognized that Hashimoto and Kilian are directed to the same basic architecture: a controller within a storage system that receives commands and policy-type instructions from a host and, based on those instructions, manages how content is stored, protected, and retrieved on behalf of the host. Ex. 1006, [0082]-[0084] (“The information processing system 1 functions as a storage system that writes to or reads data from

the storage device 2....The storage devices 2 are semiconductor storage devices configured to write data to and read data from a nonvolatile memory thereof.”), [0096]-[0097] (“The host 3 (host software) can manage and control the storage devices 2 by transmitting various commands to the storage device 2. In the present embodiment, an advanced application interface...enables the host 3 to manage and control the storage devices 2.”), [0107]-[0110] (“In the present embodiment, the storage device 2 is the SSD including the NAND flash memory (hereinafter called flash memory)....The storage device 2 includes a controller 14, a RAM 15, a flash memory 16....The RAM 15 includes a storage region to store a lookup table (LUT) 19.”); Ex. 1007, 3:16-22 (“For example, the storage system may store data (write) in response to a request from the host computer..., return stored data (read) in response to a request from the host computer..., and delete stored data in response to a request from the host computer to delete the data.”), 3:40-63 (“Storage system 101 may be accessed by hosts 105 for the purposes of storing, retrieving, or performing operations on data stored on storage system 101.”).

222. Kilian’s retention period policy would have made Hashimoto’s system more reliable. Hashimoto describes an SSD that already accepts host-supplied settings controlling how data is stored and protected. Ex. 1006, [0189]-[0195]. Notably, Hashimoto provides hosts with the ability to implement settings prioritizing data reliability. *Id.*, [0521]-[0524]. Kilian supplies an additional, known host-

supplied protection setting that improves reliability—preventing deletion or modification during a retention period. This is especially beneficial given Kilian’s policy improves reliability of read-only data by preventing corruption or malicious overwrites by an attacker. Ex. 1007, 1:38-42.

B. Reasonable Expectation of Success

223. A person of ordinary skill in the art would have had a reasonable expectation of success in making this combination. Hashimoto’s controller 14 already receives and executes a variety of host-supplied storage device policies and stores the resulting state (*e.g.*, tier attributes, compression flags) in its own onboard memory for use in subsequent read and write operations. Ex. 1006, [0110] (“The lookup table (LUT) 19 stores mapping information indicating relationship between logical addresses and physical addresses of the flash memory 16.”), [0150]-[0155] (“The controller 14 includes a CPU 61, a RAM interface 62, a NAND interface 63, and at least one hardware engine 64....The NAND interface is a NAND controller configured to control the flash memory 63 under control of the CPU 61.”), [0189]-[0195] (describing “in-drive tiering control”), [0217]-[0227] (describing “data compression control”).

224. Controller 14 is already configured to receive a host-supplied policy, store it, and consult it when executing subsequent commands targeting specific data. Therefore, adding a further host-supplied policy directing the controller to deny

deletion or modification requests for a designated period (as taught by Kilian, Ex. 1007, 3:23-39) requires nothing more than programming the controller with one additional rule of the same general kind it already implements (*i.e.*, recording a retention date in the LUT 19). A person of ordinary skill in the art would have had the knowledge and experience to make such a programming modification.

225. Such a person would have had a reasonable expectation of success in combining Kilian's retention/write-protection policy with Hashimoto's controller, because the combination uses each reference for its own established purpose and does not require any new hardware, new host-device interface, or unpredictable modification to Hashimoto's existing architecture. Rather, Kilian's policy would simply direct Hashimoto's controller 14 to check an additional host-supplied parameter (a retention period stored in the LUT) before executing a delete or overwrite operation. This would be done in the same manner that controller 14 in Hashimoto already checks other host-supplied parameters (*e.g.*, compression attribute) before executing write and read operations.

C. Dependent Claims 7, 15

1. Claim 7: “The apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

226. In my opinion, Hashimoto in view of Kilian renders obvious claim 7: “[t]he apparatus of claim 1, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

227. Hashimoto describes a *device controller* (see Section X.A.3), a *storage request* (see Section X.A.5), and a *content identifier* (see Section X.A.7). In my opinion, to the extent that Hashimoto does not explicitly disclose *wherein the content identifier is assigned by the device controller and is returned in a response to the storage request*, this would have been obvious in view of Kilian.

228. Kilian explains that the storage system, rather than the host, generates the identifier used to reference stored content. After generation, the storage system communicates that identifier back to the host so the host can use it to access the data afterward: “the content address is generated by the storage system itself, and is returned to the host.” *Id.*, 9:41-43; *see also id.*, 4:65-5:11 (“Alternatively, when the host sends data to the storage system, only the storage system may compute the content address and return the computed content address to the host for later use in accessing the data.”). This content address is generated by the controller within the storage system as part of processing the host’s storage request, and the returned

address is what the host subsequently uses “for use in the future to retrieve the data from the storage system.” *Id.*, 4:65-5:1.

229. A person of ordinary skill in the art would have been motivated to modify Hashimoto’s controller 14 to assign its own content identifier and return it to the host in the write command’s completion response. This would strengthen the security of the system, rather than allowing Hashimoto’s controller 14 to merely rely on the host to supply the Starting LBA identifying the write data (Ex. 1006, [0121])), by taking advantage of Kilian’s expressly-taught benefit of a trusted, storage-system-authenticated identifier that a host cannot spoof to circumvent the storage device’s data-management policies. Ex. 1007, 9:46-56 (generating and returning the content address thereby “ensuring that retention policies are met, irrespective of the behavior of hosts connected thereto”). A person of ordinary skill in the art would have had a reasonable expectation of success in the combination because it only requires relocating a known identifier-assignment function from the host to a controller already responsible for maintaining an identifier-to-physical address mapping in the LUT 19 and providing this identifier to the host when confirming command completion. This would have been within the knowledge and experience of a person of ordinary skill in the art at the time.

2. Claim 15: “The apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

230. In my opinion, Hashimoto in view of Kilian renders obvious claim 15: “[t]he apparatus of claim 9, wherein the content identifier is assigned by the device controller and is returned in a response to the storage request.”

231. The combination of Hashimoto and Kilian discloses the limitations of claim 15. Claim 15 depends from claim 9, and recites limitations identical to those of claim 7. Therefore, a person of ordinary skill in the art would have been motivated to combine these references, for at least the reasons above for claim 7. *See* Section X.A.1.

D. Independent Claim 16

1. Preamble 16[pre]: “An apparatus comprising:”

232. I have been asked to assume the preamble of claim 16 is limiting. Under that assumption, it is my opinion that Hashimoto discloses the preamble of claim 16 as I discussed for 1[pre]/9[pre] in Ground 3. *See* Section X.A.1.

2. Limitation 16[a]: “a storage media;”

233. In my opinion, Hashimoto discloses 16[a]: “a storage media.”

234. Hashimoto describes that storage device 2 (*apparatus*) includes flash memory 16 (*a storage media*) as I discussed for 1[a]/9[a] in Ground 3. *See* Section X.A.2. Flash memory 16 is a form of storage media.

3. Limitation 16[b]: “a device controller operatively coupled with the storage media configured to at least:”

235. In my opinion, Hashimoto discloses 16[b]: “a device controller operatively coupled with the storage media configured to at least.”

236. Hashimoto explains that storage device 2 (*apparatus*) also includes controller 14, which is electrically connected via a flash memory interface 21 (*operatively coupled with*) flash memory 16 (*a storage media*) as I discussed for 1[b]/9[b] in Ground 3. *See* Section X.A.3. The electrical connection between these two components would render the components *operatively coupled*, because data could pass between the two.

4. Limitation 16[b.i]: “receive, from a host, a storage policy, wherein the storage policy comprises at least a write-protection policy and recording method for writing content to the storage media;”

237. In my opinion, Hashimoto in view of Kilian renders obvious 16[b.i]: “receive, from a host, a storage policy, wherein the storage policy comprises at least a write-protection policy and recording method for writing content to the storage media.”

238. As I discussed for 1[b.i]/9[b.i] in Ground 3, Hashimoto explains that controller 14 (*device controller*) receives host-supplied settings from **host 3** (*receive, from a host, at least one storage device policy*). These settings include a tier attribute, which further determines a SLC, MLC, or other writing method to write

data 41 (*comprises at least a...recording method for writing content*) to the flash memory 16 (*storage media*). See Section X.4. The tier attribute also determines another type of retention, data retention. Ex. 1006, [0519] (“host 3 to designate the tier attribute giving a higher priority to the write speed than to data retention or the tier attribute giving a higher priority to the data retention than to the write speed.”). To the extent that Hashimoto does not explicitly disclose settings that prevent data from being overwritten (*a write-protection policy*), it is my opinion that this would have been obvious to a person of ordinary skill in the art in view of Kilian.

239. The '023 patent explains that write-protection determines whether data is deleted:

A user sends a request to store content in step 965. At a minimum, this request comprises content to be stored. In an implementation, the user may include information such as an identification code, encoding information, or ***data protection information***, for example. The device controller 130 uses the storage device policy to determine...how to store the content. In an implementation, this determination includes a determination that the content should be stored as ***write-protected (or write-once)*** such that i[t] will ***not be deleted*** later or overwritten.

Ex. 1001, 22:29-43.

240. Like Hashimoto, Kilian describes host-supplied settings sent to a data storage system. Kilian explains that its host provides “a retention period that indicates a period of time for which the data may not be deleted.” Ex. 1007,

Abstract. In response, the controller enforces this policy from the host when it “receives a request to delete the data, it first evaluates the retention period ... [and i]f the retention period has not expired, the storage system denies the request.” *Id.* Therefore, the controller executes actions based on the policy that it is being provided by the host. In the specific example above, the host-supplied retention period is a *write-protection policy* because it instructs the controller to take no further action and protect content from deletion until the period elapses.

5. Limitation 16[b.ii]: “receive, from the host, a storage request to store content;”

241. In my opinion, Hashimoto discloses 16[b.ii]: “receive, from the host, a storage request to store content.”

242. As I discussed for 1[b.ii] in Ground 3, Hashimoto describes that controller 14 (*device controller*) receives write command 40 (*a storage request*) from **host 3** (*from the host*) to store data 41 (*to store content*). See Section X.A.5.

6. Limitation 16[b.iii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy;”

243. Hashimoto discloses 16[b.iii]: “store the content on the storage media by utilizing the recording method indicated by the at least one storage device policy.”

244. As I discussed for 1[b.iii]/9[b.ii] in Ground 3, Hashimoto explains that controller 14 (*device controller*) writes the data (*store the content*) on flash memory 16 (*on the storage media*) based on the writing method indicated by the tier attribute

designated in the host-supplied settings (*indicated by the at least one storage device policy*). See Section X.A.6.

7. Limitation 16[b.iv]: “record storage information for the content on a memory of the device controller, comprising at least a content identifier for the content;”

245. Hashimoto in view of Kilian renders obvious this limitation.

246. As I discussed for 1[b.iv]/9[b.iii] in Ground 3, Hashimoto explains that controller 14 (*device controller*) stores (*record*) mapping and processing information associated with data 41 (*storage information for the content*) in LUT 19 in RAM 15 of the device controller (*a memory of the device controller*), which includes the LBA for the data 41 (*comprising at least a content identifier for the content*). See Section X.A.7.

247. In combining Hashimoto and Kilian, the processing information associated with data 41 would include the retention period policy as taught by Kilian in LUT 19. Kilian teaches that the data store system receives “a retention period that indicates a period of time for which the data may not be deleted.” Ex. 1007, Abstract. Therefore, “[w]hen the storage system receives a request to delete the data, it first evaluates the retention period associated with that data to determine if the retention period has expired.” *Id.*

LUT 19

Logical addresses (LBA)	Physical addresses	Compression flags	Others

FIG. 61

Ex. 1006, Fig. 61 (annotations in color)

248. In my opinion, a person of ordinary skill in the art would have been motivated to include the retention period in LUT 19 because it would centralize the processing information associated with the data located at the LBA. Hashimoto explains that LUT 19 includes other processing information about the data located at the LBA, and it would have been a simple modification of the code to include the retention period as taught by Kilian, providing a person of ordinary skill in the art with a reasonable expectation of success. Ex. 1006, Fig. 61 (as annotated above); Ex. 1007, Fig. 4.

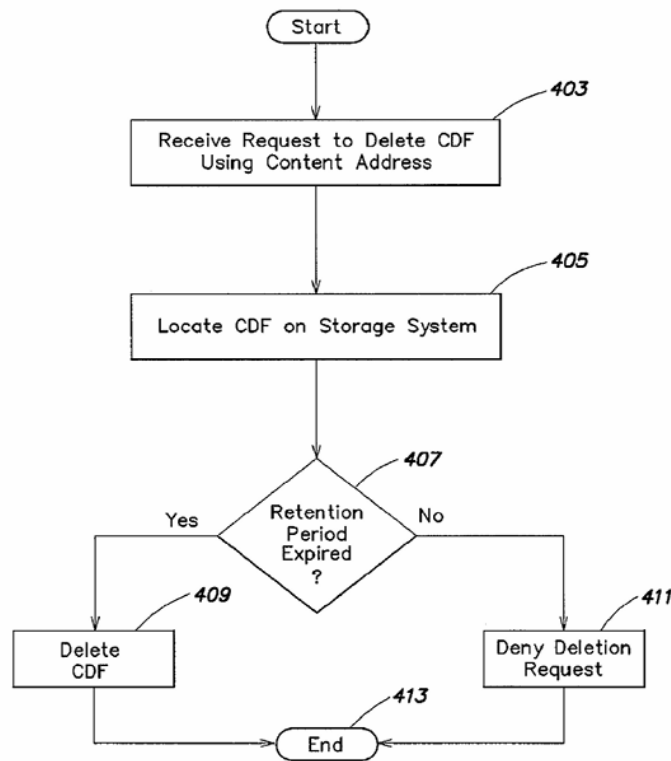


FIG. 4

Ex. 1007, Fig. 4

8. **Limitation 16[b.v]:** “subsequent to storing the content, receive a delete request for the content, comprising at least the content identifier for the content;”

249. In my opinion, Hashimoto in view of Kilian renders obvious 16[b.v]: “subsequent to storing the content, receive a delete request for the content, comprising at least the content identifier for the content.”

250. As I discussed for 1[b.v]/9[b.iv], Hashimoto explains that *subsequent to writing data 41 (storing the content)*, controller 14 (*device controller*) can receive

commands related to data 41 (*a request for the content*), which includes the LBA for data 41 (*comprising at least the content identifier for the content*). See Section X.A.8.

251. To the extent Hashimoto does not describe receiving a delete command for data 41 (*receive a delete request for the content*), this would have been rendered obvious by the Hashimoto-Kilian system. As I discussed above for 16[b.i], Kilian details that the data storage system “receives a request to delete the data” (*receive a delete request*). Ex. 1007, Abstract; *see also* Section XI.A.4. In the combination of Hashimoto and Kilian, such a delete request would be for the previously-stored data 41.

9. Limitation 16[b.vi]: “in response to the delete request, retrieve from the memory of the device controller the storage information for the content; and”

252. In my opinion, Hashimoto in view of Kilian renders obvious 16[b.vi]: “in response to the delete request, retrieve from the memory of the device controller the storage information for the content.”

253. As I discussed for 1[b.vi]/9[b.v] in Ground 3, Hashimoto describes that *in response to* a command (*request*), controller 14 (*device controller*) *retrieve[s]* from LUT 19 in RAM 15 of controller 14 (*memory of the device controller*) the L2P mapping and processing information for the data referenced in the command (*storage information for the content*). See Section X.A.9.

254. In the Hashimoto-Kilian system, the L2P mapping and processing information, including the retention period, would also be retrieved in response to the delete command (*in response to the delete request*). Ex. 1007, Abstract, 3:31-34 (“If the storage system later receives a request from a host computer to delete or modify the data, the storage system may first evaluate the stored retention period associated with the data to determine if the retention period has expired.”); Fig. 4.

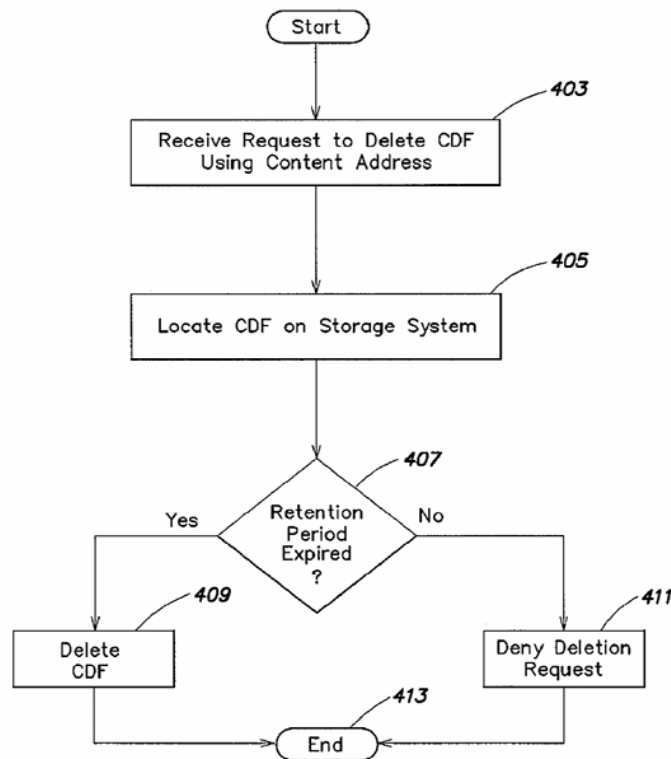


FIG. 4

Ex. 1007, Fig. 4

10. Limitation 16[b.vii]: “respond to the delete request, indicating that the content was not deleted in accordance with the write-protection policy.”

255. In my opinion, Hashimoto in view of Kilian renders obvious 16[b.vii]: “respond to the delete request, indicating that the content was not deleted in accordance with the write-protection policy.”

256. Hashimoto explains that the controller 14 (*device controller*) will respond to commands (*respond to the...request*). As Hashimoto shows in Fig. 3, the storage device 2 through the controller 14 will send a reply to the host whenever an instruction or request is received. Ex. 1006, [0102]-[0104] (“The file system 12 transmits the command to the storage device 2. When a response from the storage device 2 is received, the file system 12 transmits the response to the OS 11. The[n], the OS 11 transmits the response to the application software layer 13.”), Fig. 3.

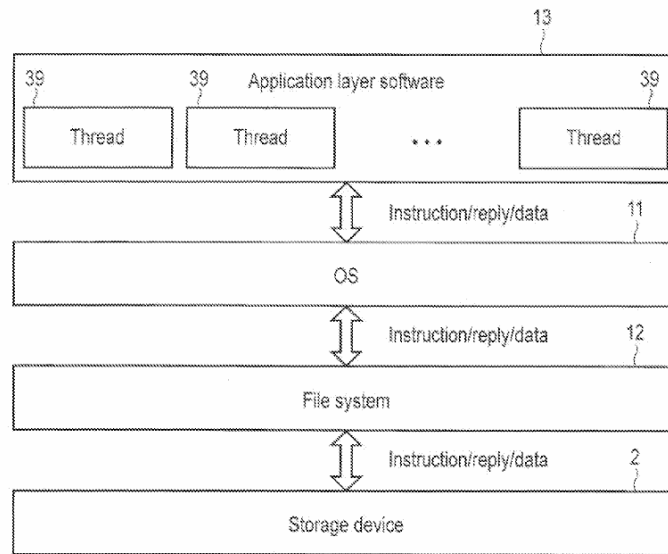


FIG. 3

Ex. 1006, Fig. 3

257. When combined with Kilian's system, Hashimoto's controller 14 responds to the delete command (*respond to the delete request*), *indicating that data 41 was not deleted due to the retention period (the content was not deleted in accordance with the write-protection policy)*. Kilian explains that its controller makes a determination regarding the delete command when it "receives a request to delete the data," evaluates the retention period, and denies the request if "the retention period has not expired." Ex. 1007, 2:5-10. In the combination, Hashimoto's controller 14 retrieves the storage information for the targeted content from the LUT 19 and evaluates Kilian's retention period. *Id.*, 2:6-8 ("The at least one controller receives a request, from the host, to delete a unit of data stored on the

storage system”), 10:8-10 (“The process then proceeds to act 407, where the process examines the metadata of the CDF and evaluates the retention period to determine if the retention period has expired.”). The controller denies the delete request if the period has not expired, responding to the host by indicating that the content was not deleted. *Id.*, 10:8-20 (“When it is determined... that the retention period has not yet expired, [the] process denies the request to delete the CDF,” with “[t]he storage system ... optionally return[ing] an error message to the requester (e.g., a host).”).

258. Therefore, by returning the error message to the requester, the controller *indicates that the content was not deleted in accordance with the write-protection policy*, which prevented the delete request from being completed.

E. Dependent Claims 17-20

1. Claim 17: “The apparatus of claim 16, wherein the storage policy and the delete request are received from the same source.”

259. In my opinion, Hashimoto in view of Kilian renders obvious claim 17: “[t]he apparatus of claim 16, wherein the storage policy and the delete request are received from the same source.”

260. To the extent that Hashimoto does not explicitly disclose *wherein the storage policy and the delete request are received from the same source*, in my opinion this would have been obvious in view of Kilian. As I explained above for

element 1[b.i]/9[b.i], Hashimoto's storage device policy is received from the host.
See Section X.A.4.

261. Kilian's delete request is similarly received from the host. Ex. 1007, 1:59-60, 2:6-7 (the storage system deletes stored data "in response to a request from the host computer to delete the data"). Because the host computer directs the storage system to delete data, this is a delete request that is received from the host computer. Therefore, in the Hashimoto-Kilian combination, both the storage policy and the delete request are thus received from the host.

2. Claim 18: "The apparatus of claim 16, wherein the storage policy and the delete request are received from different sources."

262. In my opinion, Hashimoto in view of Kilian renders obvious claim 18: "[t]he apparatus of claim 16, wherein the storage policy and the delete request are received from different sources."

263. To the extent that Hashimoto does not explicitly disclose *wherein the storage policy and the delete request are received from different sources*, it is my opinion that this would have been obvious in view of Kilian. Kilian describes a storage system that communicates with multiple different sources. Ex. 1007, 3:42-43 ("communicates with a plurality of hosts 105a, 105b, ..., 105m through a network 103."). Kilian explains that the storage system enforces retention policies, regardless of the connected host. *Id.*, 9:50-54 ("In this respect, one embodiment of the present

invention is directed to providing a storage system that can meet with rigorous requirements for ensuring that retention policies are met, irrespective of the behavior of hosts connected thereto.”). In the Hashimoto-Kilian combination, the storage policy is received from one host and a delete request may be received from any one of a different host, satisfying this limitation.

3. Claim 19: “The apparatus of claim 16, wherein the delete request is a modification request.”

264. In my opinion, Hashimoto in view of Kilian renders obvious claim 19: “[t]he apparatus of claim 16, wherein the delete request is a modification request.”

265. To the extent that Hashimoto does not explicitly disclose that *the delete request is a modification request*, in my opinion this would have been obvious in view of Kilian. Kilian’s retention-period policy expressly includes modification. Kilian defines the retention period as “a period of time for which the data cannot be deleted or modified.” Ex. 1007, 3:23-29. Kilian further confirms that the storage system evaluates its retention policy against both types of requests: “If the storage system later receives a request from a host computer to delete or modify the data, the storage system may first evaluate the stored retention period associated with the data to determine if the retention period has expired.” *Id.*, 3:31-35. Additionally, Kilian explains a process for modifying data such that the content addresses are consistent, which further confirms the support that Kilian provides. *Id.*, 5:25-31. The storage system thus receives and denies modification requests in the same

manner as delete requests when the retention period has not expired. In the Hashimoto-Kilian combination, Hashimoto's controller enforces Kilian's retention policy against modification requests (*delete request that is a modification request*).

4. Claim 20: "The apparatus of claim 16, wherein the delete request is an overwrite request."

266. In my opinion, Hashimoto in view of Kilian renders obvious claim 20: "[t]he apparatus of claim 16, wherein the delete request is an overwrite request."

267. To the extent that Hashimoto does not explicitly disclose that *the delete request is an overwrite request*, in my opinion this would have been obvious in view of Kilian. Kilian explains that traditionally, storing modified data could "overwrite the original data," and Kilian's system prevents this by ensuring that modified data "is stored at a different location from original data 201 and does not overwrite the original data." Ex. 1007, 5:33-36. A person of ordinary skill in the art would see that an overwrite is a form of a modification request, because the overwrite would change the data at that location. Further, the L2P contains a single unique mapping; an overwrite of data changes that mapping so that old data is no longer accessible, so that is in effect, deleted.

268. Kilian's write-protection policy thus treats an *overwrite request* (i.e., a request to write new data to the location of previously stored content) as a form of deletion request that is denied when the retention period has not expired. In order for the deletion request to be granted, Kilian's write-protection policy must ensure

that the retention period has expired. Therefore, in the Hashimoto-Kilian combination, Hashimoto's controller enforces Kilian's retention policy against overwrite requests (*delete request that is a overwrite request*).

XII. OPINIONS ON GROUND 2, CONTINUED: CLAIMS 32, 34, AND 36 ARE OBVIOUS IN VIEW OF OLSTER IN COMBINATION WITH DAVIS

A. Claim 32

1. Preamble 32[pre]: "An apparatus comprising:"

269. I have been asked to assume the preamble of claim 16 is limiting. Under that assumption, it is my opinion that Olster discloses the preamble of claim 16 as I discussed for 1[pre]/9[pre] in Ground 1. *See* Section VIII.A.1.

2. Limitation 32[a]: "a storage media;"

270. In my opinion, Olster discloses 32[a]: "a storage media."

271. Olster describes that storage 150 (*apparatus*) includes storage media 160 (*a storage media*) as I discussed for 1[a]/9[a] in Ground 1. *See* Section VIII.A.2.

3. Limitation 32[b]: "a device controller operatively coupled with the storage media configured to at least:"

272. In my opinion, Olster discloses 32[b]: "a device controller operatively coupled with the storage media configured to at least."

273. Olster explains that storage 150 (*apparatus*) also includes a controller for a node, which may communicate with and allow for communications with

(*operatively coupled with*) storage media 160 (*a storage media*) as I discussed for 1[b]/9[b] in Ground 1. *See* Section VIII.A.3.

4. Limitation 32[b.i]: “receive, from a host, a storage device policy, wherein the storage device policy indicates recording methods for writing content to the storage media wherein each recording method corresponds to a content type;”

274. In my opinion, Olster in view of Davis renders obvious 32[b.i]: “receive, from a host, a storage device policy, wherein the storage device policy indicates recording methods for writing content to the storage media wherein each recording method corresponds to a content type.”

275. Olster discloses *receiving, from a host, a storage device policy, wherein the storage device policy indicates recording methods for writing content to the storage media* as I discussed for 1[b]/9[b] in Ground 1. *See* Section VIII.A.4.

276. To the extent that Olster does not expressly describe a *recording method for writing content to the storage media*, Olster in view of Davis renders obvious this limitation as I discussed in Ground 2. *See* Section IX.C.

277. To the extent that Olster does not expressly describe a *wherein each recording method corresponds to a content type*, Olster in view of Davis renders obvious this limitation as I discussed for claim 2 in Ground 2. *See* Section IX.D.1. For example, Davis teaches differentiating between text data and image data by assigning a method of writing to lower density type memory to text data and

assigning a method of writing to higher density type memory to image data. Ex. 1008, 9:37-45 (“a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell”); *see* Section IX.D.1. Therefore, the data properties (the content type) determine the method of writing, such as the density type of the memory. *See* Section IX.D.1.

5. Limitation 32[b.ii]: “identify the content type of a first content and the content type of a second content;”

278. In my opinion, Olster in view of Davis renders obvious 32[b.ii]: “identify the content type of a first content and the content type of a second content.”

279. To the extent that Olster does not expressly describe a *identify the content type of a first content and the content type of a second content*, Olster in view of Davis renders obvious this limitation as I discussed for claim 2 in Ground 2. *See* Section IX.D.1. For example, Davis teaches identifying text data as a first content type and image data as a second content type. Ex. 1008, 9:37-45; *see* Section IX.D.1.

6. Limitation 32[b.iii]: “store the first content on the storage media by utilizing a first recording method of the recording methods in accordance with the storage device policy; and”

280. In my opinion, Olster in view of Davis renders obvious 32[b.iii]: “store the first content on the storage media by utilizing a first recording method of the recording methods in accordance with the storage device policy.”

281. Olster discloses *store the ... content on the storage media by utilizing a ... recording method ... in accordance with the storage device policy* as I discussed for 1[b.iii]/9[b.ii] in Ground 1. *See* Section VIII.A.6.

282. To the extent that Olster does not expressly describe a *store the **first** content on the storage media by utilizing a **first** recording method of the recording methods in accordance with the storage device policy*, Olster in view of Davis renders obvious this limitation as I discussed for claim 2 in Ground 2. *See* Section IX.D.1. For example, Davis teaches assigning a method of writing to lower density type memory (*first recording method*) to text data (*the first content*). Ex. 1008, 9:37-45 (“a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell”); *see* Section IX.D.1.

7. Limitation 32[b.iv]: “store the second content on the storage media by utilizing a second recording method of the recording methods in accordance with the storage device policy wherein the first density recording method and the second recording method differ.”

283. In my opinion, Olster in view of Davis renders obvious 32[b.iv]: “store the second content on the storage media by utilizing a second recording method of the recording methods in accordance with the storage device policy wherein the first density recording method and the second recording method differ.”

284. Olster discloses *store the ... content on the storage media by utilizing a ... recording method ... in accordance with the storage device policy* as I discussed for 1[b.iii]/9[b.ii] in Ground 1. *See* Section VIII.A.6.

285. To the extent that Olster does not expressly describe a *store the **second** content on the storage media by utilizing a **second** recording method of the recording methods in accordance with the storage device policy wherein the first density recording method and the second recording method differ*, Olster in view of Davis renders obvious this limitation as I discussed for claim 2 in Ground 2. *See* Section IX.D.1. For example, Davis teaches assigning a method of writing to higher density type memory (*second recording method*) to image data (*the second content*). Ex. 1008, 9:37-45 (“a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell”); *see* Section IX.D.1. Davis further teaches that the two recording methods differ – one is writing to higher density type memory, and the other is writing to lower density type memory. *Id.*

B. Dependent Claims 34, 36

8. Limitation Claim 34: “The apparatus of claim 32, wherein the first content and the second content are both stored on the storage media.”

286. In my opinion, Olster in view of Davis renders obvious claim 34: “The apparatus of claim 32, wherein the first content and the second content are both stored on the storage media.”

287. Olster discloses that the “[s]torage media include magnetic media...or flash memory.” EX1005, [0029]. Davis teaches that “the flash memory 132 includes a single level cell type of flash memory 210, [and] a multilevel cell type of flash memory 212.” EX1008, 4:39-40. In the combination resulting from these teachings, the text data (*first content*) and the image data (*second content*) would have been stored in the SLC memory 210 and MLC memory 212, respectively, in the Olster’s storage media 160 (*both stored on the storage media*).

9. Limitation Claim 36: “The apparatus of claim 34, wherein the first content and the second content are different content types.”

288. In my opinion, Olster in view of Davis renders obvious claim 36: “The apparatus of claim 34, wherein the first content and the second content are different content types.”

289. As discussed for claim 32, the text data (*first content*) and image data (*second content*) are different types of data with different characteristics (*content type*). EX1008, 9:39-43 (“This policy could be interpreted as...a policy of writing text data to a type of flash memory having a lower number of bits per cell and writing image data to a type of flash memory having a higher number of bits per cell.”); *see* Section XII.A. For example, text data and image data have different sensitivities to errors.

XII. CONCLUSION

290. For all the reasons stated above, it is my opinion that the challenged claims of the '023 patent are unpatentable.

291. This declaration and my opinions are made to the best of my knowledge and understanding, and based on the material available to me at the time of signing this declaration. All statements made herein of my own knowledge are true, and all statements made on information and belief are believed to be true. Further, I am aware that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001. I declare under penalty of perjury that the foregoing is true and correct.

Declaration of Heiner Litz, Ph.D. in Support of
Petition for *Inter Partes* Review USP No. 10,776,023

Executed this 05 day of 08 2026 in Heidelberg, Germany.