IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Micron Technology, Inc.; Micron Semiconductor Products, Inc.; and Micron Technology Texas LLC,

Petitioners,

v.

Unification Technologies LLC,

Patent Owner.

Case No. IPR2021-00941

U.S. Patent No. 8,762,658

PETITION FOR *INTER PARTES* REVIEW OF CLAIMS 1-5 AND 8-12 OF U.S. PATENT NO. 8,762,658

TABLE OF CONTENTS

Page

TABLE OF AUTHORITIES

Cases **Page(s)**

Statutes

Other Authorities

PETITIONERS' EXHIBIT LIST

Ex. No.	Brief Description
1001	U.S. Pat. No. 8,762,658 B2, titled "SYSTEMS AND METHODS FOR PERSISTENT DEALLOCATION," to Flynn et al.
1002	U.S. Provisional Pat. App. No. 60/912,728, titled "REMOVE-ON-DELETE TECHNOLOGIES FOR SOLID STATE DRIVE OPTIMIZATION," to Frank Shu et al. ("Shu Provisional").
1003	U.S. Pat. No. 9,207,876, titled "REMOVE-ON-DELETE TECHNOLOGIES FOR SOLID STATE DRIVE OPTIMIZATION," to Frank Shu et al. ("Shu Patent").
1004	Expert Declaration of Jacob Baker, Ph.D., P.E., Regarding U.S. Patent No. 8,762,658 (June 4, 2021).
1005	American National Standard for Information Technology–ATA/ATAPI Command Set – 2 (ACS-2), ANSI INCITS 482-2012 (May 30, 2012) (excerpts filed with permission).
1006	Serial ATA: High Speed Serialized AT Attachment Revision 1.0, Serial ATA International Organization (Aug. 29, 2001).
1007	Serial ATA (SATA) ATA Revision 2.5, Serial ATA International Organization (Oct. 27, 2005).
1008	William D. Brown & Joe E. Brewer, *Nonvolatile Semiconductor Memory Technology* (IEEE 1998).
1009	Brian Dipert & Markus Levy, *Designing with FLASH MEMORY* (Annabooks 1994).
1010	April 2007 Plenary Minutes 2, e07156r0 (T13 and INCITS, Apr. 24, 2007).
1011	U.S. Pat. No. 8,533,406 B2, titled "APPARATUS, SYSTEM, AND METHOD FOR IDENTIFYING DATA THAT IS NO LONGER IN USE," to Flynn et al.
1012	Original Complaint for Patent Infringement, *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020), ECF No. 1.
1013	Exhibit A to Plaintiff's First Amended Infringement Contentions: Unification Technologies' Allegations of Infringement with Respect to U.S. Patent No. 8,762,658, *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020).

1014	Frank Shu & Nathan Obr, *Data Set Management Commands Proposal for ATA8-ACS2*, T13 (rev. 6, Dec. 12, 2007).
1015	U.S. Provisional Pat. App. No. 60/873,111, titled "ELEMENTAL BLADE SYSTEM," to Flynn et al. (Dec. 6, 2006) ("2006 Provisional").
1016	Docket Report for *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020) (accessed June 3, 2021).
1017	Frank Shu, *Notification of Deleted Data Proposal for ATA8-ACS2*, T13 (rev. 0, Apr. 21, 2007).
1018	Frank Shu & Nathan Obr, *Data Set Management Commands Proposal for ATA8-ACS2*, T13 (rev. 1, July 26, 2007).
1019	Excerpt of Plaintiff's Responses to Defendant's First Set of Interrogatories (No. 16) (April 26, 2021), *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020).
1020	Eran Gal et al., *Mapping Structures for Flash Memories: Techniques and Open Problems*, PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON SOFTWARE—SCIENCE, TECHNOLOGY & ENGINEERING (digital version), Herzlia, Israel, 2005, pp. 83-92, doi: 10.1109/SWSTE.2005.14.
1021	Public file history of U.S. Pat. No. 8,672,658, titled "SYSTEMS AND METHODS FOR PERSISTENT DEALLOCATION," to Flynn et al.
1022	Frank Shu, Solid-State Drives: Next-Generation Storage, Microsoft WinHEC 2007 (May 14-17, 2007).
1023	Wayback Machine Archive of Microsoft WinHEC 2007 Conference Presentations Website, captured Sept. 12, 2007, https://web.archive.org/web/20070214023104/http://www.microsoft.com/whdc/winhec.
1024	Frank Shu, Windows 7 Enhancements for Solid-State Drives, Microsoft WinHEC 2008 (Nov. 4-6, 2008).
1025	U.S. Pat. No. 5,404,485A, titled "FLASH FILE SYSTEM," to Ban.
1026	Curriculum vitae of Jacob Baker, Ph.D., P.E.
1027	H. Niijima, *Design of a Solid-State File Using Flash EEPROM*, IBM JOURNAL OF RESEARCH AND DEVELOPMENT, vol. 39, no. 5, pp. 531-45, Sept. 1995.
1028	U.S. Pat. No. 7,057,942, titled "MEMORY MANAGEMENT DEVICE AND MEMORY DEVICE" to Suda et al.

1029	U.S. Pat. No. 7,624,239, titled "METHODS FOR THE MANAGEMENT OF ERASE OPERATIONS IN NON-VOLATILE MEMORIES," to Bennett et al.
1030	Plaintiff's Reply Claim Construction Brief, *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020), ECF No. 57.
1031	Computer-generated redline comparison of changes from the Shu Provisional to the Shu Patent.
1032	SD Specifications Part 1 PHYSICAL LAYER Simplified Specification Version 1.10, SD Group and SD Card Association Technical Committee (Mar. 18, 2005).
1033	U.S. Pat. 6,014,724, titled "FLASH TRANSLATION LAYER BLOCK INDICATION MAP REVISION SYSTEM AND METHOD," to Jenett.
1034	U.S. Provisional Pat. App. No. 60/974,470 for "Apparatus, System, and Method for Object-Oriented Solid-State Storage," to Flynn et al. ("September 2007 Provisional").
1035	Claim Construction Order, *Unification Techs. LLC v. Micron Tech. Inc.*, No. 6:20-cv-500 (W.D. Tex. 2020), ECF No. 67.
1036	U.S. Pat. No. 6,677,432 B2, titled "MEMORY MANAGEMENT SYSTEM SUPPORTING OBJECT DELETION IN NON-VOLATILE MEMORY," to Saltz et al.
1037	U.S. Pat. No. 9,632,727 B2, titled "SYSTEMS AND METHODS FOR IDENTIFYING STORAGE RESOURCES THAT ARE NOT IN USE," to Flynn et al.
1038	Expert Report of Sylvia D. Hall-Ellis, Ph.D., Regarding Public Availability of the IBM Journal Publication (Ex. 1027), dated May 26, 2021.
1039	Declaration of Frank Shu Regarding Publication of Proposals (Exs. 1017, 1018), dated June 2, 2021.
1040	American National Standard for Information Technology—AT Attachment with Packet Interface – 7 Volume 1 – Register Delivered Command Set, Logical Register Set (ATA/ATAPI-7 V1), ANSI INCITS 397-2005 (Feb. 7, 2005) (excerpts filed with permission).

I. Introduction

The challenged claims in the U.S. Patent No. 8,762,658 ("the '658 Patent") should never have issued. The fatal flaw for these claims arises from the classic scenario of an applicant forced to amend claims to overcome prior art rejections, in a manner unsupported by the broad, high-level disclosure of an earlier provisional application. When applying the proper priority date—here, the September 22, 2007 filing date of the second provisional application[1] (the "September 2007 Provisional")—intervening prior art invalidates the challenged claims.

The intervening prior art in this case is exceptionally strong. In co-pending litigation, Unification Technologies LLC ("UTL") alleges that challenged claims cover products that implement the "Trim" command. Here, the intervening prior art consists of Frank Shu's initial proposal of the accused Trim command (Ex. 1017) to the standards-setting Technical Committee 13 ("T13") and his U.S. Pat. No. 9,207,876 encompassing the same. Ex. 1003. Indeed, Frank Shu led Microsoft's efforts to announce that the popular Windows operating system would support Trim

[1] Petitioners assume priority for the challenged claims is properly supported by the September 2007 Provisional application because the prior art herein predates September 22, 2007, but Petitioners reserve the right to challenge this assumption in the co-pending district court litigation.

months before the assumed September 2007 priority date for the challenged claims. Ex. 1022 at 1, 2, 9; Ex. 1023 at 3; Ex. 1024 at 1, 2, 5.

Given these facts, the Board can easily resolve the present IPR challenge. If the Board finds that the earliest provisional application does not fully support the challenged claims, the claims are unquestionably invalid.

II. Petitioners Meet Requirements for *Inter Partes* Review

Petitioners certify under 37 C.F.R. § 42.104(a) that the '658 Patent "is available for *inter partes* review and that the Petitioners are not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in the petition." UTL sued Petitioners for alleged infringement less than one year ago on June 5, 2020. Exs. 1012, 1016.

III. Prosecution History of the '658 Patent

During examination, to overcome a rejection based on U.S. Patent No. 6,014,724 to Jenett, the applicant amended claim 51 (later issued as claim 1) to recite: "a message comprising a logical identifier, the message indicating ~~an indication~~ that data associated with [[a]] the logical identifier has been erased." Ex. 1021 at 1234. The applicant argued that in light of this amendment, the prior art does not comprise a message compromising a "logical identifier [that] has been erased." *Id.* at 1241-42. Section VI below explains why the earliest 2006 provisional application for the '658 Patent ("the 2006 Provisional") lacks support

for this amendment.

On September 25, 2013, the applicant disclosed a December 12, 2007 proposal by Frank Shu to T13 in an information disclosure statement ("the 2013 IDS") as one of 191 references. *Id.* at 980 (cite no. D19). The 2013 IDS, however, failed to indicate that the proposal was the "sixth revision" of Frank Shu's proposal or otherwise suggest the possibility of earlier revisions that predate the September 2007 Provisional. Ex. 1014, cover. The Examiner likely realized that the September 2007 Provisional predated the cited December 12, 2007 date without further considering the reference and without realizing that earlier proposals existed.

The applicant's amendment and selective IDS disclosure of the sixth Trim proposal led the Examiner to believe that "the art of record fails to teach or suggest receiving a message/hint/identifier comprising a logical identifier that indicates data associated with the logical identifier has been erase[d]." Ex. 1021 at 1261-63.

IV. Technology Background

Flash memory is a form of solid-state nonvolatile computer memory. Flash memory is organized in erasable units called "blocks," which are made up of smaller "pages." Ex. 1004 (Expert Declaration of Jacob Baker, hereinafter "Baker") ¶ 110.

Since at least the early 1990s, the generic architecture of both flash (e.g., solid-state drive ("SSD")) and magnetic-platter (e.g. hard disk drive ("HDD")) mass-storage devices has included: (1) an interface, (2) a controller to manage data in the

storage device, and (3) a storage medium in the form of flash memory or a magnetic platter. *E.g.*, Ex. 1009 at 66, Fig. 4.14 (reproduced below); Ex. 1025 at Fig. 1.

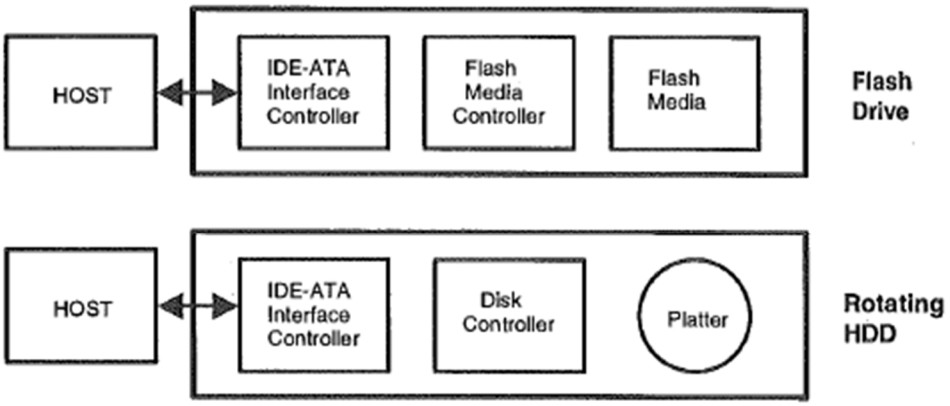


Figure 4.14: Mass Storage Architecture

Flash memory has long used a flash translation layer ("FTL") to map logical addresses to physical addresses. Baker ¶¶ 128-129; *see also, e.g.*, Ex. 1020 at 3, 9 (crediting Ban with patenting the FTL in 1995); Ex. 1025 (Ban's FTL patent); Ex. 1033, 5:21-26; Ex. 1015 at 77 (admitting that "[t]raditional" flash storage systems used FTL's like those patented by M-Systems, referring to Ex. 1025); Ex. 1027 Fig. 1 (showing direct and reverse Address Translation Tables between logical addresses ("LA") and physical addresses ("PA")). The FTL allows computer systems to operate and address data in a logical address space (e.g., logical address 0x0000 through 0xFFFF) without concern for where a solid-state storage device physically saves the data (e.g., in which particular block/page). Baker ¶ 131.

Unlike magnetic-platter hard drives, flash memory cannot be directly overwritten—a block must be erased before written to again. *Id.* ¶ 72. To improve

an SSD's ability to identify and erase invalid data, Jenett invented sending file indication maps from an operating system to a flash memory controller so that the flash memory controller can track invalid blocks and erase them. Ex. 1033. Later, Frank Shu improved upon Jenett's idea and proposed that the operating system instead send the Trim command, which specifies the logical block addresses of invalid data, instead of sending an entire file indication map. Exs. 1003, 1017. Frank Shu then led Microsoft Windows to announce support for the Trim command. Ex. 1022 at 8; Ex. 1023; Ex. 1024 at 2, 4, 10. The industry would go on to adopt the Trim command as part of the ACS-2 standard. Ex. 1005 § 7.9.3.2.

V. Summary of the '658 Patent

Independent claim 1 recites an apparatus that includes a non-volatile storage medium, a request receiver module, and a storage module. Ex. 1001, 53:2-16. The request receiver module is configured to receive a message comprising a logical identifier that is mapped to a physical storage location of the non-volatile storage medium. *Id.* The message indicates that "data associated with the logical identifier has been erased." *Id.* The storage module is configured to store persistent data on the non-volatile storage medium where the persistent data is (1) configured to indicate that the data associated with the logical identifier is erased and (2) stored in response to the indication that data associated with the logical identifier has been erased. *Id.*

UTL contends that the "message" in claim 1 covers Frank Shu's Trim command. Ex. 1013, *passim*.

VI. The Priority Date of the '658 Patent Does Not Precede September 22, 2007

The '658 Patent claims priority to provisional application nos. 60/873,111, filed December 6, 2006, and 60/974,470, filed September 22, 2007. The disclosure of the '658 Patent diverged greatly from the 2006 Provisional. In fact, none of the inventors executed declarations contemporaneous with the as-filed '658 Patent application. Instead, the applicant recycled declarations signed five years prior to the '658 Patent's 2012 filing date. Ex. 1021 at 152-54.

The 2006 Provisional does not support the priority date for the challenged claims. Instead, as shown herein, the 2006 Provisional lacks support for at least three claim limitations of issued claim 1:

(1) "a message comprising a logical identifier, the message indicating that data associated with the logical identifier has been erased,"

(2) "a storage module configured to store persistent data on the non-volatile storage medium in response to the indication," and

(3) "the persistent data is configured to indicate that the data associated with the logical identifier is erased."

The 2006 Provisional also fails to support the dependent claims. UTL contends that certain sections of the 2006 Provisional provide support (Ex. 1019 at

35-36), but each section lacks support for the reasons explained below. Thus, the priority date for the challenged claims comes on or after September 22, 2007.

A. Priority Requires Every Limitation to Have Explicit, Implicit, or Inherent Support

To comply with the written description requirement of 35 U.S.C. § 112 and receive an earlier priority date under 35 U.S.C. § 120,[2] each claim limitation must be expressly, implicitly, or inherently supported in the originally filed disclosure. MPEP § 2163(II)(A)(3)(b). That original disclosure "must describe the invention sufficiently to convey to a person of skill in the art that the patentee had possession of the claimed invention at the time of the application, i.e., that the patentee invented what is claimed." *LizardTech, Inc. v. Earth Res. Mapping, Inc.*, 424 F.3d 1336, 1345 (Fed. Cir. 2005). "In other words, the specification of the *provisional* must 'contain a written description of the invention and the manner and process of making and using it, in such full, clear, concise, and exact terms,' 35 U.S.C. § 112 ¶ 1, to enable an ordinary skilled artisan to practice the invention *claimed* in the *non-provisional* application." *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). One skilled in the art, reading the original disclosure, "must immediately discern the limitation at issue in the claims." *Purdue Pharma L.P. v.*

[2] Nothing in 35 U.S.C. § 112 or § 120 allows the Patent Owner to provide support by arguing that the claims were obvious under § 103 in view of the provisional.

Faulding Inc., 230 F.3d 1320, 1323 (Fed. Cir. 2000).

The written description requirement "guards against the inventor's overreaching by insisting that he recount his invention in such detail that his future claims can be determined to be encompassed within his original creation." *Vas-Cath Inc. v. Mahurkar*, 935 F.2d 1555, 1561 (Fed. Cir. 1991). If the originally filed disclosure does not provide support for each claim limitation, a priority or benefit claim under 35 U.S.C. § 120 must be denied. MPEP § 2163(II)(A)(3)(b).

Narrowing the claims by introducing elements or limitations that are not supported by the as-filed disclosure is a violation of the written description requirement of pre-AIA 35 U.S.C. § 112. MPEP § 2163.05(II); *see, e.g.*, *Rozbicki v. Chiang*, 590 F. App'x 990, 996 (Fed. Cir. 2014) (nonprecedential) (finding that patentee "cannot now improperly narrow its language by importing limitations not supported by the claim language or written description").

When an explicit limitation in a claim "is not present in the written description whose benefit is sought[,] it must be shown that a person of ordinary skill would have understood, at the time the patent application was filed, that the description *requires* that limitation." MPEP § 2163(II)(A)(3)(b) (citing cases) (emphasis added). "Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d

1949, 1950-51 (Fed. Cir. 1999) (citations omitted). A subgenus is not necessarily implicitly described by a genus encompassing it and a species upon which it reads. *In re Smith*, 458 F.2d 1389, 1395, 173 USPQ 679, 683 (CCPA 1972). "[O]ne cannot disclose a forest in the original application, and then later pick a tree out of the forest and say here is my invention. In order to satisfy the written description requirement, the blaze marks directing the skilled artisan to that tree must be in the originally filed disclosure." *Purdue Pharma*, 230 F.3d at 1326-27.

When filing an amendment to pending claims, "an applicant should show support in the original disclosure for new or amended claims." MPEP § 2163(II)(A)(3)(b); *see also* MPEP §§ 714.02 and 2163.06 ("Applicant should ... specifically point out the support for any amendments made to the disclosure."). The applicant here never argued that the claim amendments at issue are supported by the 2006 Provisional. Ex. 1021 at 1232-43.

B. Summary of the 2006 Provisional

The 2006 Provisional contains a scattershot of separate ideas catalogued as "claims," none of which match the '658 Patent's challenged claims. *Compare* Ex. 1015, *with* Ex. 1001. Most of these "claims" in the 2006 Provisional have nothing to do with the challenged claims. *See, e.g.*, Ex. 1015, 20-22 (describing "identical card pairings" for pairing together cards in a blade server chassis), 24 (describing a business model of leasing storage to third parties).

UTL asserts in the co-pending litigation that the 2006 Provisional's "claim" 18 provides § 112 support for the challenged claims of the '658 Patent. Ex. 1030 at 10 (citing the "empty block directive").

"Claim" 18 of the 2006 Provisional presents a high-level problem and solution. The problem: "Garbage collection based storage systems such as those commonly used with NAND flash get very poor performance when they do not have enough free space." Ex. 1015 at 40. The stated solution references an "empty-block directive" or "hint":

> ***Statement of Solution***
>
> **Most agents that use block storage do not need the contents of every block to be preserved. File systems, for example, are rarely filled to near the capacity of the block storage on which they are storing the data. If the file system were to supply a hint to the block storage regarding which specific blocks do not hold data that needs to be preserved, the efficiency of the garbage collection on the underlying block storage system can be greatly enhanced.**
>
> **The empty-block directive can be added to the block storage API and protocols. File systems and other clients of that API/protocol can be enhanced to issue these directives. For example, when a file is deleted, the file-system can issue an "empty-block" directive for the blocks that contained the data for that file, but no longer need to remember the contents. This directive could even serve a secondary security purpose by incorporating a flag to indicate that the data not only need not be preserved, but should be destroyed.**

Id. The applicant also provided an "alternative" solution. In the alternative solution, instead of sending the hint/command, the 2006 Provisional states that agents can write zeros to the storage medium, thereby overwriting the blocks whose contents are no longer needed:

> An alternative to introducing this empty-block directive could be to have the agents simply write all zeros to the blocks whose contents are no longer needed. The underlying block storage system can recognize all-zero blocks and avoid having to actually store the contents. Subsequent reads can return the same all-zero data. Certain file systems such as XFS, for security purposes, have options for zeroing out the blocks that held data from deleted files. Simply enabling those options could be sufficient to allow garbage collection based block storage systems to achieve high efficiency by interpreting those zeroed blocks as "free". Or, those file systems could be enhanced to issue an "empty-block" directive in place of zeroing the blocks.

Id.

C. The 2006 Provisional Has No Support for Claim 1's "message comprising a logical identifier, the message indicating that data associated with the logical identifier has been erased"

As explained in Section IV above, Jenett invented sending a file indication map to an SSD to indicate invalid data, and in April 2007, Frank Shu proposed that this type of command should identify the specific logical block address of invalid data (instead of sending a whole file indication map). Ex. 1033; Exs. 1017-1018. The 2006 Provisional lacks any disclosure of this concept.

Instead, the 2006 Provisional has no disclosure about the structure of the "hint" or "empty-block" directive other than it could incorporate "a flag" that indicates whether data should be destroyed. Baker ¶¶ 68-69. Nowhere does the 2006 Provisional state that the "empty-block" hint/directive should comprise "a logical identifier." *Id.* ¶¶ 68-87. At best, the 2006 Provisional describes a hint "regarding which specific blocks do not hold data" and that "file-systems can issue an 'empty-block' directive for the blocks that contained data for that file." Ex. 1015 at 40. But this disclosure falls short of disclosing that the hint or directive uses or "compris[es] a logical identifier" for identifying such blocks, as opposed to

identifying the blocks some other way. For these reasons, a POSITA would not have understood "claim" 18 of the 2006 Provisional to explicitly teach that the "indication comprises a logical identifier." Baker ¶¶ 68-69. Thus, the 2006 Provisional contains no explicit support of a "message comprising a logical identifier" as recited in the challenged claims. *Id*.

Nor does the 2006 Provisional provide an inherent or implicit disclosure of the empty-block hint/directive comprising a logical identifier. *Id*. ¶¶ 69-72. In a related patent, the applicant claimed the empty-block directive separately from logical block addresses. Ex. 1037, 54:55-56. The '658 Patent discloses that some commands use physical addresses, not logical addresses, to identify blocks. Ex. 1001, 15:29-32, 18:18-26, 21:47-53, 35:44-48. Based on this disclosure, a POSITA would have recognized that a hint or directive might have identified a physical address instead of the claimed logical identifier. Baker ¶ 70. Indeed, a POSITA would have recognized even more alternatives for the structure of the hint or directive. For example, a different instruction, separate from the hint or directive, could include the logical identifier, which is similar to how erase commands worked in the SD Specification. *Id.*; Ex. 1032 § 4.3.5 (requiring, in sequence, a command CMD32 to identify the START block, a separate command CMD33 to identify the END block, and finally the command CMD38 to ERASE the previously indicated blocks). As yet another example, Jenett solved this same problem by sending an

entire "file indication map." Ex. 1033, Fig. 5 block 600, *passim*; Ex. 1021 at 936-37.

Because the empty-block hint/directive does not comprise a logical identifier, the 2006 Provisional further lacks support for "data associated with the logical identifier has been erased." The 2006 Provisional describes providing a "hint" that "specific blocks do not hold data that needs to be preserved" and further describes "when a file is deleted, the file system can issue an 'empty-block directive' for the blocks that contained the data for that file, but no longer need to remember the contents." Ex. 1015 at 40. The 2006 Provisional further describes potentially "incorporating a flag to indicate that data not only need not be preserved, but should be destroyed." *Id.*

But the descriptions of data that "do not … need[] to be preserved" or "should be destroyed" does not support claim 1's recital of "data associated with the logical identifier has been erased." Baker ¶ 71. Instead, these disclosures teach that the data has not been erased: either the data is still present and (i) need not be "preserved" or (ii) the data is still present and "should be destroyed." *Id.* ¶¶ 71-72. In fact, these disclosures would have led a POSITA away from a message indicating that data "has been erased." *Id.* ¶ 71. Thus, the empty-block hint/directive has nothing to do with indicating a message using a logical identifier to indicate that anything associated with the logical identifier "has been erased," as claimed in the

'658 Patent. *Id.* ¶¶ 71-72.

The 2006 Provisional mentions that one example condition for sending an empty-block directive is when "a file is deleted." Ex. 1015 at 40. But the 2006 Provisional does not say that the empty-block directive indicates that the file is erased and that the file is identified using a logical identifier. The language describing what is indicated is: "for the blocks that contained the data for that file, but no longer need to remember the contents." *Id.*; Baker ¶ 72. This means that the data exists and no longer needs to be remembered; the data is not indicated as "erased" as claimed. Baker ¶ 72.

For these reasons, "claim" 18 of the 2006 Provisional fails to expressly, implicitly, or inherently disclose any of these options, much less the specific option that the hint or directive comprises "a logical identifier."

UTL cannot salvage support from other parts of the 2006 Provisional. The alternative solution of writing zeros also makes no reference to using a logical identifier. Ex. 1015 at 40. Aside from "claim" 18, the only other 2006 Provisional "claims" that mention an empty-block hint/directive include "claims" 14 and 29. Baker ¶¶ 75, 82; Ex. 1015 at 35, 103. But, these "claims" discuss entirely different problems and solutions, and neither discloses a hint/directive comprises a logical identifier. Baker ¶¶ 75, 82; Ex. 1015 at 35, 103. Moreover, provisional "claim" 14 operates under an incompatible assumption—"that every block contains contents

that must always be remembered—in other words, no block can be considered free space". Ex. 1015 at 35; Baker ¶ 73. If anything, provisional "claim" 14, titled "object based storage with garbage collection," shows that the inventors had not solved how to integrate object storage and garbage collection because this provisional "claim" recites incomplete and incomprehensible sentences as the purported "solution." Ex. 1015 at 35; Baker ¶ 75.

Furthermore, other passages in the 2006 Provisional mention a "logical block" in other contexts, but these parts also fail to describe an empty-block hint/directive involving a logical identifier. For example, "claims" 22 and 24 of the 2006 Provisional mention logical blocks *in a map* but say nothing about receiving logical identifiers with the empty-block hint/directive. Baker ¶ 74; Ex. 1015 at 68, 72-74.

Additionally, to the extent that UTL refers to the 2006 Provisional's "claim" 30, this description only involves an ObjectID as part of object-based commands—a type of command different from block commands such as the empty-block hint/directive. Baker ¶ 83; Ex. 1015 at 104. Also, this provisional "claim" directs the POSITA away from block-based commands entirely because it purportedly gives the POSITA "a killer reason to change from block access to object access." Ex. 1015 at 104; Baker ¶ 83. Provisional "claims" 31 and 32 also discuss an incompatible object-based system and provide no further detail about the empty-block hint/directive. Ex. 1015 at 105-06; Baker ¶ 83.

Provisional "claims" 22 and 23 mention "messages," but these "messages" refer to "units on NAND flash" or, in other words, "encapsulate raw data" on a media. Ex. 1015 at 69-70. "A message is written by placing commands on the command queues." *Id.* at 69. Thus, the disclosures about the structure of a "message" in provisional "claims" 22 and 23 do not support the structure of a command like the empty-block hint/directive. Baker ¶¶ 77-78.

Provisional "claim" 27(E) relates to a "NAND controller" and a "NAND Write Agent." Ex. 1015 at 92. This "claim" relates to a controller command for writing new data to pages of flash memory and has nothing to do with the empty-block hint/directive sent from an operating system to the SSD. Baker ¶ 80.

The remaining provisional "claims" 2 and 9 cited by UTL have nothing to do with the empty-block hint/directive. Ex. 1015 at 23, 31; Baker ¶ 86.

For these reasons, the 2006 Provisional fails to describe "a message comprising a logical identifier" in sufficient detail so that one skilled in the art could reasonably conclude that the applicant had possession of the claimed invention and thus fails to provide written description support for the challenged claims. Baker ¶ 87.

D. The 2006 Provisional Has No Support for the "storage module configured to store persistent data in the non-volatile storage medium in response to the indication"

The 2006 Provisional makes no mention of a storage module "configured to

store persistent data on the non-volatile storage medium in response to the indication." Ex. 1001, 53:11-13; Ex 1015; Baker ¶¶ 89-101.

"Claim" 18 in the 2006 Provisional only discloses what the empty-block hint/directive signifies, not what to do afterward—especially not what a *storage module* does in response to the empty-block hint/directive. Ex. 1015 at 40; Baker ¶¶ 90-92. Moreover, nothing in the 2006 Provisional discloses storing any type of *persistent* data in response to the empty-block hint/directive. Baker ¶ 93. The lack of this detail is important because the 2006 Provisional teaches that some data structures "may not be stored persistently at all." Ex. 1015 at 68.

At best, the 2006 Provisional only mentions that as a result of the hint/directive, "the efficiency of the garbage collection on the underlying block storage system can be greatly enhanced." Ex. 1015 at 40. But the 2006 Provisional never explains how. Baker ¶ 92. The lack of this teaching leaves the implementation to the imagination of a POSITA, including whether to preserve the data or destroy it, as signified by the empty-block hint/directive. *Id.* ¶¶ 91, 94. In any event, the 2006 Provisional's disclosure fails to suggest the possibility of storing more, persistent data in response to the empty-block hint/directive. *Id*. Indeed, continuation Patent No. 8,533,406 describes different actions optionally performed in response to receiving an indication. Ex. 1011, 54:15-55:14. For these reasons, the 2006 Provisional fails to expressly, implicitly, or inherently disclose a "storage

module configured to store persistent data in the non-volatile storage medium in response to the indication." Baker ¶¶ 89-94.

UTL cannot salvage support from other parts of the 2006 Provisional. For example, provisional "claim" 25 describes garbage collection that operates under the opposite, incompatible assumption: "Blocks are always considered valid, their contents needing to be preserved, even if the client (say a file system) doesn't have anything useful stored in a given block." Ex. 1015 at 74; Baker ¶ 96. This incompatible assumption would have prevented a POSITA from combining provisional "claim" 18's "hint ... regarding which specific blocks do not hold data that needs to be preserved" with the garbage collection system of provisional "claim" 25. Ex. 1015 at 40, 74; Baker ¶ 96. Provisional "claim" 25 also describes a different solution that occurs in response to a different condition: "Whenever data is appended to the medium, we identify the old data that is becoming garbage." Ex. 1015 at 75. This disclosure of identifying old data in response to a different condition (when new data is appended) does not support the '658 Patent's claimed storage module configured to store persistent data indicating that the data associated with the logical identifier in the message is erased in response to the indication. Baker ¶ 97.

Along the same lines, provisional "claim" 29 relates to "Object Storage" and only briefly mentions "providing an 'empty-block' directive can additionally

improve the efficiency of emulating a block device on top of object based storage that uses garbage collection underneath" without the further detail needed to support the claims. *Id.* ¶ 98; Ex. 1015 at 103. Provisional "claim" 29 says nothing about what happens during garbage collection in response to receiving the empty-block hint/directive. Baker ¶ 98.

Aside from provisional "claims" 18 and 29, only provisional "claim" 14 mentions the same empty-block hint/directive. Claim 14 discloses nothing about storing persistent data in response to the empty-block hint/directive and provides an incomprehensible solution. *Id.*; Ex. 1015 at 35. Also, provisional "claim" 24 relates to identifying "bad" blocks that are "no longer useable" due to defects, but this has nothing to do with the empty-block hint/directive. Ex. 1015 at 72; Baker ¶ 99. None of the other provisional "claims" 2, 9, 22-25, 27, or 30-32 further describe any persistent storage in response to the empty-block hint/directive for the reasons discussed in section VI.C above.

For these reasons, the 2006 Provisional fails to describe a "storage module configured to store persistent data in the non-volatile storage medium in response to the indication" in sufficient detail so that a POSITA could reasonably conclude that the applicants had possession of the claimed invention and thus fails to provide written description support for the challenged claims. Baker ¶¶ 88-101.

E. The 2006 Provisional Has No Support for "the persistent data is configured to indicate that the data associated with the logical identifier is erased"

For the reasons discussed above, the 2006 Provisional does not disclose what the claimed message indicates and what a storage module does in response to receiving the message. For the same reasons, the 2006 Provisional fails to further disclose, in response to receiving a message comprising a logical identifier, storing persistent data that indicates that the data associated with the logical identifier "is erased."

The 2006 Provisional discloses that the empty-block hint/directive "regard[s] which specific blocks do not hold data that needs to be preserved" and "could even . . . incorporat[e] a flag to indicate that data not only need not be preserved, but should be destroyed." Ex. 1015 at 40. These sentences teach that the data "should be destroyed" or "not preserved," meaning that the data is *not yet erased*. Baker ¶ 103. Along the same lines, the 2006 Provisional discloses, "For example, when a file is deleted, the file-system can issue an 'empty-block' directive for the blocks that contained the data for that file, but no longer need to remember the contents." Ex. 1015 at 40. This sentence has nothing to do with storing persistent data indicating that data is erased because "no longer need to remember" means that the data still exists at this time but may be erased in the future. Baker ¶ 103. Thus, these sentences would have led a POSITA *away* from using persistent data to "indicate

that the data associated with the logical identifier is erased." *Id.* If anything, this provisional disclosure would have led a POSITA to understand that associated data *does not need to be preserved*, or perhaps that data *should be destroyed*, but certainly not that the data "is erased," as claimed. *Id*.

For these reasons, the 2006 Provisional fails to describe "the persistent data is configured to indicate that the data associated with the logical identifier is erased" in sufficient detail so that one skilled in the art could reasonably conclude that the applicants had possession of the claimed invention and thus fails to provide written description support for the challenged claims. *Id.* ¶¶ 102-104.

F. The Dependent Claims Similarly Lack Priority

Claims 2-5 and 8-12 depend from claim 1 and thus lack priority support from the 2006 Provisional because of dependency.

Claims 2-5 independently lack priority support from the 2006 Provisional because the 2006 Provisional contains no details about the claimed "reconstruction module" at all, much less a reconstruction module that indicates the various types of "erased" information recited in claims 2-5. *Id*. ¶ 106. Claim 8 independently lacks priority because the 2006 Provisional says nothing about the claimed "read request module" returning the recited indication while data remains on the physical storage location. Ex. 1015 at 40, 70, 96; Baker ¶ 106. Claims 9-10 independently lack priority because the 2006 Provisional says nothing about the claimed "marking

module" that operates as claimed in response to an indication. Ex. 1015 at 40; Baker ¶ 106. Claims 11-12 independently lack priority because the 2006 Provisional says nothing about the claimed "storage recovery module" or "erase module" that operates in response to the indication. Ex. 1015 at 40; Baker ¶ 106.

UTL repeatedly cites provisional "claims" 30-32 to support the challenged dependent claims. Ex. 1019 at 35-36. Not only do these provisional "claims" 30-32 fail to support the dependent claims, these provisional "claims" relate only to an incompatible object storage system instead of a block-based system. Baker ¶¶ 73, 106; Ex. 1015 at 104-06. Aside from provisional "claim" 18, other provisional "claims" assume that "the contents of every block must be preserved ALWAYS," which is incompatible with an empty-block hint/directive that indicates certain data does not need to be preserved. Ex. 1015 at 35 ("Block based storage assumes that every block contains contents that must always be remembered."), 104; Baker ¶¶ 73, 106. UTL also relies on the 2007 Provisional disclosure of read agents and erase agents, but these sections lack disclosure about the claimed functionality and say nothing about operating in response to the empty-block hint/directive. Ex. 1015 at 94, 96.

Thus, the 2006 Provisional disclosure would not have conveyed possession of these dependent claims to a POSITA, and the priority claim to the 2006 Provisional "must be denied" for the dependent claims. Baker ¶ 107; MPEP § 2163(II)(A)(3)(b).

VII. Level of Ordinary Skill in the Art

A POSITA in September 2007 would have a bachelor of science degree in computer science or electrical engineering and at least two years of experience in the design, development, implementation, or management of solid-state memory devices. Baker ¶ 58. The references cited in this Petition, the state of the art, and the experience of Dr. Jacob Baker as described in his expert declaration (Ex. 1004) reflect this level of skill in the art. In this Petition, reference to a POSITA refers to a person with these or similar qualifications.

The POSITA in September 2007 would have also known about Frank Shu's Trim proposals to T13 and Microsoft's announcement to support Trim at WinHEC 2007. *Id.* ¶ 60. The POSITA would also have known fundamental concepts related to flash memory management. *Id.* ¶ 59 (explaining concepts).

VIII. Claim Construction

The Board construes claims under the same claim construction standard as civil actions in federal district court. The district court for the co-pending litigation has construed certain terms. Ex. 1035. Although the parties have disputed the claim constructions, the construction disputes do not affect the outcome of this Petition.[3] The district court's constructions for the claims at issue in this Petition are as follows:

[3] Petitioners reserve all rights to appeal the district court's claim constructions.

Claim Term	Court
"logical identifier" Claims 1, 4, 8-10	An identifier that can be associated with a physical address on a storage device for identifying data stored at the physical address.
"marking module" Claims 9, 10	Not indefinite; not subject to § 112(f); plain and ordinary meaning.
"storage module" Claim 1	Not indefinite; not subject to § 112(f); plain and ordinary meaning.
"data associated with the logical identifier [has been/is] erased" Claims 1, 3-5, 8	Not indefinite; plain and ordinary meaning.

UTL appears to contend that the plain and ordinary meaning of "data associated with the logical identifier [has been/is] erased" refers to "data associated with the logical identifier has been erased by a device connected to the [accused] Product (e.g., by a computer)" and, as a result, "[f]rom the user's perspective, this data has been deleted from a document." Ex. 1013 at 2. Further, UTL contends that this term does ***not*** refer to data on a non-volatile device being already physically erased. *See, e.g.*, *id.* ("The TRIM command tells the SSD that specific areas contain data ….").

IX. Precise Relief Requested

A. Ground 1

Claims 1-5 and 8-12 are rendered obvious by U.S. Pat. No. 9,207,876 to Frank Shu et al., (the "Shu Patent," Ex. 1003) in view of Frank Shu's Trim Proposals revisions 0-1 (Exs. 1017-1018), and the IBM Journal article, *Design of a solid-state file using flash EEPROM* (Ex. 1027, "IBM").

B. Qualifying Prior Art

For the reasons discussed in Section VI above, the challenged claims of the '658 Patent have an effective filing date no earlier than September 22, 2007.

The Shu Patent has priority to the "Shu Provisional," application no. 60/912,728, filed on April 19, 2007, and is § 102(a) and (e) prior art. Ex. 1003; Ex. 1002. The Shu Patent contains minor edits in comparison to its provisional. Ex. 1031 (showing computer-generated comparison). As shown in the table, the Shu Provisional provides full support for the claims under 35 U.S.C. §§ 112, 120. Thus, the Shu Patent has priority to its provisional filing date.

Shu Patent Element	**Shu Provisional Support (Ex. 1002)**
1. A system comprising:	Fig. 3 (showing system).
[1a] a computing device that includes at least one processor and memory;	Fig. 3 (showing computing device 301, processing unit 307, memory 309), [0028].
[1b] a file system; and	Fig. 1 (showing file system 112b), [0014], [0016]-[0017].
[1c] a solid state drive	Fig. 1 (showing SSD driver 114b), [0012]-[0014],

("SSD") driver that, based on execution by the at least one processor, is configured to:	[0016]-[0019], [0021], [0023], [0025]-[0027]. "Processor 307 typically processes or executes various computer-executable instructions to control the operation of computing device 301," which includes the operating system shown in Fig. 3, which includes the SSD driver shown in Fig. 1. [0026].
[1d1] receive, from a file system, a remove-on-delete command that includes invalid data information	Fig. 1 (showing SSD driver 114b, receiving the remove-on-delete command via interface 140, from the file system 112b), [0017] ("File system 112b utilizes new interface 140 to communicate invalid data information to SSD driver 114b."), [0023] ("Block 230 indicates a remove-on-delete command. This command typically includes the invalid data information and instructs the SSD device and/or its driver to mark the indicated data as invalid.").
[1d2] that indicates that, based on a deletion of at least a portion of a file in the file system, particular data that is stored on an SSD and corresponds to the at least the portion of the file is, as indicated by the deletion, considered invalid by the file system; and	[0015] ("For example, when a file is deleted, the data associated with the file is invalid."); [0017] ("File system 112b utilizes new interface 140 to communicate invalid data information to SSD driver 114b. . . . Interface 140 enables file system 112b to indicate to SSD driver 114b via the invalid data information exactly which data stored on SSD 130 are invalid."); [0014] (generally); Fig. 2 at 210-230; [0021] ("Block 210 indicates a delete event impacting data stored on the SSD device. One example of such a delete event is a file delete operation performed by a file system wherein the file is stored on an SSD device."); [0023] ("[S]uch a command is issued by the system performing the delete operation and an SSD driver.").
[1e] instruct, based on the received invalid data information, the SSD to mark the particular data invalid on the SSD.	Fig. 2 block 240, [0014], [0017], [0019] ("SSD driver 114b typically interacts with SSD 130 via interface 120b to mark appropriate data, blocks, pages, or the like as invalid."); [0020] ("Such a method may be used to mark deleted SSD data as invalid, otherwise known as 'remove-on-delete.'"); [0023] ("Block 230 indicates a remove-on-delete command. This command typically . . . instructs

	the SSD device and/or its driver to mark the indicated data as invalid."); [0024] ("Block 240 indicates marking the deleted data as invalid.").

A pre-AIA 35 U.S.C. § 102(e) prior art reference "'shall have the same effect,' including a patent-defeating effect, . . . as though it was filed on the date of the . . . provisional" to which it claims priority, as long as certain requirements are met. *In re Giacomini*, 612 F.3d 1380, 1383-84 (Fed. Cir. 2010) (quoting 35 U.S.C. § 119(e)). In particular, the Board has held that a § 102(e) reference is available as prior art as of its provisional application's filing date when the provisional provides support for: (1) at least one claim of the § 102(e) reference and (2) the subject matter on which the petitioner relies. *Cisco Sys., Inc. v. Capella Photonics, Inc.*, IPR2014-01276, Paper No. 40 at 21-22 (P.T.A.B. Feb. 17, 2016). With respect to the first prong, the provisional application must disclose an invention claimed in the § 102(e) reference "in the manner provided by the first paragraph of section 112." 35 U.S.C. § 119(e)(1); *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). Only one claim from the later-issued patent must be supported by the provisional. *See Cisco Sys*. IPR2014-01276, Paper No. 40 at 22 n.9; *Polaris Indus., Inc. v. Arctic Cat Inc*., IPR2016-01713, Paper 9, at 13 (P.T.A.B. Feb. 27, 2017).

The Shu Trim Proposals published from April 2007 (rev. 0) to August 2007 (rev. 1) and qualify as prior art under § 102(a). Ex. 1010 § 8.1.2; Ex. 1017; Ex. 1018;

Ex. 1039. T13 published the proposal on its freely accessible website T13.org, and industry representatives met to discuss Frank Shu's original Trim Proposal. Ex. 1010 § 8.1.2 (referencing the document number of Ex. 1017); Ex. 1039. Ground 1 below includes citations to the Shu Provisional to show that the provisional discloses the same technology in the Shu Patent.

IBM published by 1996 and qualifies as prior art under § 102(a) and (b). Ex. 1027; Ex. 1038 ¶¶ 40-47.

C. The Proposed Ground Is Not Cumulative or Redundant

The grounds for trial presented in this Petition are not cumulative to issues already examined during prosecution. The applicant never informed the Patent Office of Frank Shu's original April 21, 2007 Trim Proposal. The Patent Office did not know that UTL contends that the claims cover the Trim command. Because the applicant disclosed only the sixth revision of the Trim Proposal dated after both the 2006 Provisional and the September 2007 Provisional, the Examiner had no reason to look at the sixth revision of the Trim Proposal before allowing the application.

Not only did the Examiner never consider the April 21, 2007 Trim Proposal, the Examiner also never considered IBM. So although the Examiner considered U.S. Publication 2008/0263305 to Shu et al., the Examiner did not do so in view of the April 21, 2007 Trim Proposal and IBM as proposed herein.

The grounds for trial presented in this Petition are not cumulative to issues

already examined in a parallel IPR proceeding, IPR2021-00344. The parallel IPR proceeding assumed, without conceding, the December 2006 priority date and thus did not raise Shu Trim Proposals or the Shu Patent as prior art. The PTAB "recognizes that there may be circumstances in which more than one petition may be necessary, including, for example, . . . when there is a dispute about priority date requiring arguments under multiple prior art references." PTAB Consolidated Trial Practice Guide at 59, *available at* www.uspto.gov/sites/default/files/documents/tpgnov.pdf (Nov. 2019).

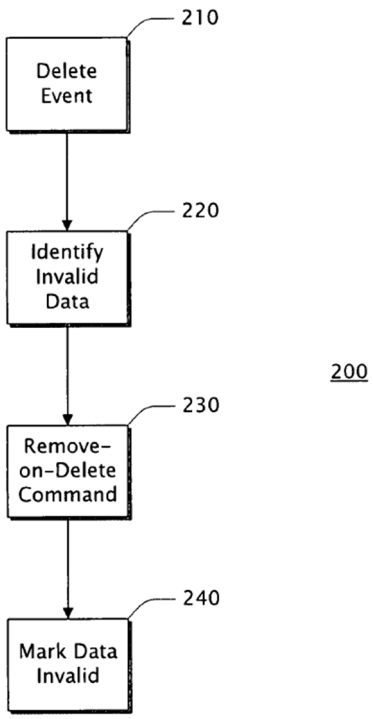
The *General Plastic* factors do not warrant denying this Petition. *Gen. Plastic Indus. Co. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (P.T.A.B. Sept. 6, 2017). IPR2021-00344 relates to overlapping claims of the '658 Patent and was filed on December 22, 2020. At that time, investigation into the Shu Trim Proposals remained ongoing, and Petitioners did not have possession of relevant materials, such as Frank Shu's presentations (Ex. 1022; Ex. 1024) until recently. UTL also delayed providing its priority contentions until late April 2021. Ex. 1019. Discovery remains ongoing. The earlier IPR petition dealt with different prior art and assumed a different priority date; thus, UTL's Preliminary Response to the earlier petition confers no unfair advantage here. The limited resources of the Board will be put to efficient use because the Board is already familiar with the technology and mainly needs to decide a priority date challenge to prevent the applicant from unfairly using

hindsight to capture Frank Shu's invention.

X. The Prior Art

A. Summary of the Shu Patent

The Shu Patent relates to managing SSDs with flash memory. Ex. 1003, 1:12-15. The Shu Patent teaches that SSDs might unnecessarily preserve invalid data during "wear leveling" and "merge" operations because SSDs "are generally unaware of what data ... is invalid." *Id.*, 1:20-43. Thus, the Shu Patent proposes that an operating system or file system send a "remove-on-delete" command to identify the invalid data to the SSD. *Id.*, 4:4-7, 4:51-5:4. The SSD can then mark the deleted data as invalid using "any form sufficient to identify the invalid data." *Id.*, 5:5-11. These invalid marks allow the SSD to avoid unnecessarily preserving the invalid data during wear leveling operations. *Id.*, abstract, 5:11-13. Shu Patent figure 2 shows the process:



B. Summary of Shu's Trim Proposals

In these Trim Proposals, Frank Shu proposes the "Trim" command, which corresponds to the "remove-on-delete" command described in the Shu Patent. Baker ¶¶ 156-157; Exs. 1017-1018. These Trim Proposals describe the Trim command that UTL now accuses of infringement. Exs. 1017-1018. The exact format of the Trim command varied across revisions, but all formats reserve bits for a logical block address ("LBA") and a "Count" field to indicate the starting address of data and a length of the data.

Word	Name	Description
00h	Feature	XXh
01h	Count	Number of 256 word-blocks of LBA Range Entry to be transferred. 0000h specifies that 65,536 blocks are to be transferred
02-04h	LBA	Reserved
05h	Command	XXh

Ex. 1017 § 3. Later revisions bundled the Trim command as part of the Data Set Management (DSM) command. Ex. 1018 § 6.1.1.

C. Summary of IBM

IBM presents "dynamic sector allocation" (also known as "wear leveling") and background garbage collection. Ex. 1027 at 531-32. IBM discusses flash EEPROM characteristics of both NOR and NAND flash memory, including the relatively slow write speeds, the need to erase "all the cells in a block … at the same time," and the inability to freely overwrite data. *Id.* at 533-34. IBM specifically addresses two problems: (1) "How to extend the lifetime" of flash memory in view of limited erase cycles, and (2) "How to develop an effective algorithm for the erase operation, so that the erase overhead will be almost hidden." *Id.* at 534.

To address both problems, IBM first uses an "*address translation table* (ATT)" to store "[t]he relation between the logical sector address and the physical sector address" in volatile memory. *Id.* at 535. The system also stores "information for reconstructing the ATT" in nonvolatile flash memory, including whether data is "valid," "invalid," or "blank." *Id.* at 535-36. The system uses the nonvolatile

information to reconstruct the ATT after each power off. *Id.*at 535.

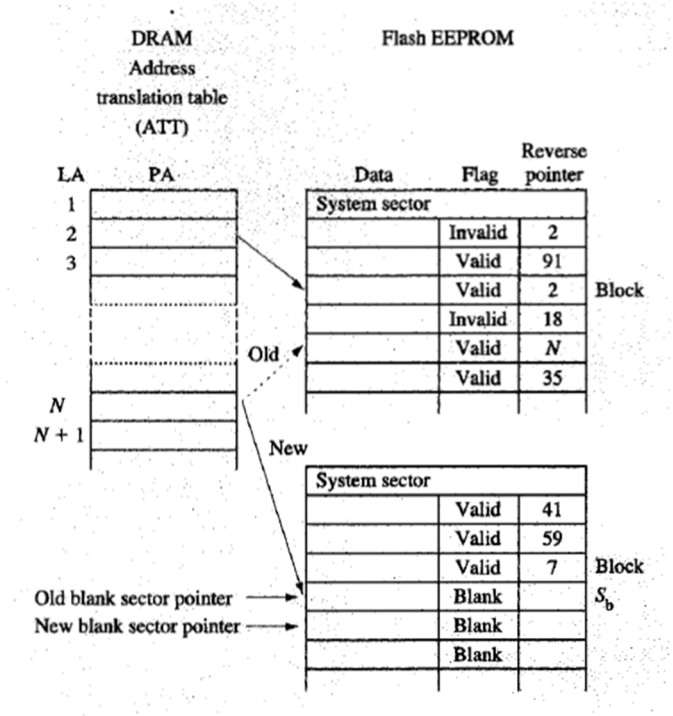


Figure 1

Id. at 535.

To avoid prematurely wearing out a block by repeatedly rewriting/erasing to the same block, a controller writes the physical addresses of a new, blank sector (e.g., S_b; sectors correspond to blocks) in the address translation table ("ATT") when writing data to the same logical address. *Id.* In the picture above, the logical address N will be associated with an address pointing to the new physical sector S_b (e.g., 4th row) instead of the old physical sector (e.g., 5th row). Baker ¶ 164. This way, the same logical address will relate to different physical addresses. *Id.*

IBM also explains the background garbage collection technique for the

erasable cluster, "which may consist of only one block." Ex. 1027 at 538-39. This technique will ultimately "[e]rase the selected cluster" that contains "garbage" (invalid data), thereby allowing the block to quickly write new data. *Id.* at 539. But before erasing the cluster, all valid sectors are copied to blank sectors, and the ATT is updated appropriately, to preserve the valid data. *Id.*

D. Motivation to Combine

The Shu Patent teaches to modify existing flash devices, such as IBM's flash memory system, to use Shu's newly proposed remove-on-delete (Trim) command. The Shu Patent implements "new functionality" by sending the new remove-on-delete (Trim) command to identify invalid data to solid-state devices. Ex. 1003, 3:62-4:9, 4:35-41, 4:65-5:4. The command could be sent to any type of SSD or flash memory device, especially those that perform wear leveling or merge operations. *Id.*, 1:20-32, 2:35-40, 2:52-67, 4:25-34.

IBM provides an example of an flash memory device that would benefit from receiving the new remove-on-delete command taught in the Shu Patent. Baker ¶ 167. Thus, a POSITA would have modified IBM's flash memory device to receive the remove-on-delete command. *Id*. IBM uses an address table for mapping logical-to-physical addresses, tracking invalid data, and erasing blocks while preserving the valid data during garbage collection. Ex. 1027 at 535-39. Frank Shu intended for his remove-on-delete command to inform flash devices like those in IBM about

which data is invalid because IBM's flash device uses "merge" operations as a part of its garbage collection process. Ex. 1003, 3:8-17; Ex. 1027 at 539 (step 3 includes merging data into another block). This additional information would improve the performance by allowing the flash memory device to avoid "unnecessarily operating on invalid data" (avoid preserving that data during garbage collection). Ex. 1003, 3:17-22, 4:29-34; Ex. 1017 § 3.

The PTAB need not speculate about a hypothetical POSITA's motivations because actual POSITAs in the solid-state industry applied the teachings of the Shu Patent to flash devices that used IBM's logical-to-physical table and garbage collection. Frank Shu proposed that flash memory representatives, including IBM's representative, use his new command. Ex. 1010 §§ 3.3, 8.1.2. Then in May 2007, Microsoft announced that the Windows operating system would support the Trim command for SSDs. Ex. 1022 at 8. POSITAs in T13 considered and adopted the command into an industry standard. Ex. 1005 § 7.9.3.2.

A POSITA would have further turned to the Shu Trim Proposals (Exs. 1017-1018) for supplemental details about the operation of the remove-on-delete/Trim command because Frank Shu submitted his Trim Proposals to an industry standard-setting organization, T13, which is responsible for the ATA standard. A POSITA would have looked to T13 submissions and wanted to comply with the ATA industry standards because the Shu Patent specified that the remove-on-delete/Trim

command would use the ATA interface. Ex 1003, 2:49-51, 4:18-20, claims 7, 12, and 18; Ex. 1002 [0012], [0018].

Thus, a POSITA would have been motivated—and POSITAs were, in fact, motivated—to add support for Shu's new command to a flash memory device like the one described in IBM. Baker ¶ 168. When doing so, POSITAs would have looked to the Shu Trim Proposals for details about how the command would be implemented in the industry standard. *Id.*

XI. Ground 1: Obvious over the Shu Patent (Ex. 1003) in View of the Shu Trim Proposals (Exs. 1017-1018) and IBM (Ex. 1027)

UTL accuses the Trim command, as used by generic SSDs, of infringing the claims. Ex. 1013, *passim*. But as explained in Section VI above, Frank Shu disclosed the Trim command to the public months before the applicant ever attempted to support the asserted claims. Thus, Frank Shu's Trim Proposals, combined with details of a generic flash memory device (e.g., as shown in IBM), invalidate the claims under UTL's interpretation.

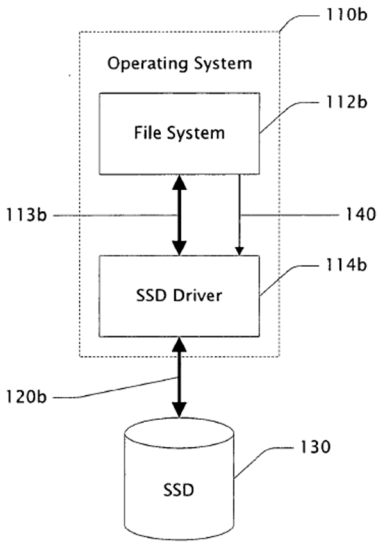
Petitioners allege in co-pending litigation that accused products that implement the TRIM command do not infringe, but for purposes of this IPR petition, Petitioners use UTL's interpretations. The Board and Federal Circuit have approved of this procedure in several matters. *See, e.g.*, *Spherix Inc. v. Matal*, 703 F. App'x 982, 983 (Fed. Cir. 2017) (approving petitioner's proposal of patent owner's claim interpretations); *Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11

at 12 (P.T.A.B. Nov. 10, 2020) ("Petitioner's alternative pleading before a district court is common practice, especially where it concerns issues outside the scope of *inter partes* review.").

A. Claim 1

a) Element 1[a][4]

If the preamble is limiting, the Shu Patent discloses the claimed apparatus in the form of a computer system with a solid-state storage device ("SSD"). Ex. 1003, Figs. 1 (cropped, reproduced below), 3 (reproduced below); Ex. 1002,[5] Figs. 1, 3; Baker ¶¶ 170-171.



[4] See appended Claim Listing.

[5] Citations to Exhibit 1002 show the Shu provisional application disclosures the relied-upon parts of the Shu Patent.

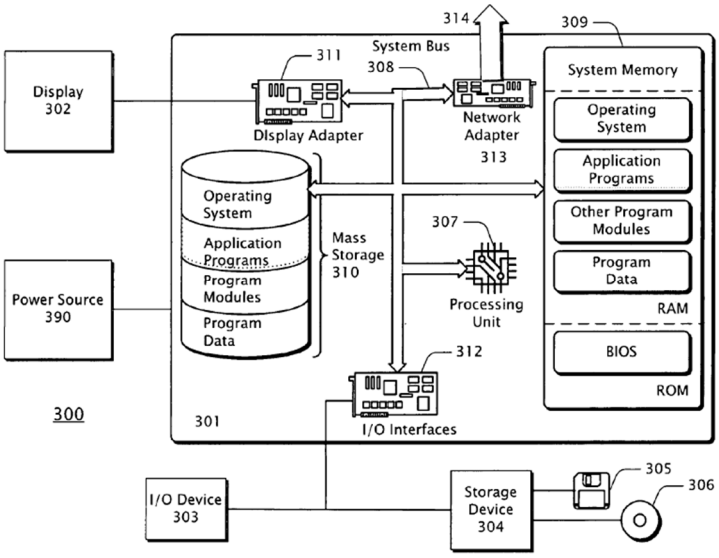


FIG. 3

b) Element 1[b]

The Shu Patent shows an SSD in figure 1 (reproduced above) and discloses, "SSDs are commonly fabricated to include flash memory devices, such as nonvolatile flash memory devices." Ex. 1003, 1:13-16; *see also id.* at 2:52-57; Ex. 1002, [0001], [0013]. Thus, a POSITA would have understood Shu's SSD in figure 1 and Shu's storage device in figure 3 to have included a nonvolatile storage medium in the form of nonvolatile flash memory. Baker ¶ 173.

c) Element 1[c]

UTL accuses the "SATA interface" of infringing this element and contends that the request receiver module refers to "circuitry, software, and/or firmware for receiving information over a SATA bus." Ex. 1013 at 2.

The Shu Patent teaches the same thing. A POSITA would have found it obvious that the Shu Patent's SSD would have the same SATA interface accused by

UTL. Baker ¶¶ 175-176. Figures 1 and 3 of the Shu Patent show an SSD interfacing with a host to receive commands, including receiving a "remove-on-delete" (Trim) command. Ex. 1003, abstract, 1:60-63, 3:20-22, 4:35-38, 4:65-5:4; Ex. 1002 at 23, [0003], [0014], [0020], [0023]. The interfaces for receiving the new remove-on-delete/Trim command "include the advanced technology attachment ('ATA') interface." Ex. 1003, 2:49-51; Ex. 1002, [0012]. A POSITA would have understood this to refer to two interfaces: the same SATA interface accused by UTL and the parallel ATA interface. Baker ¶¶ 175-176. Moreover, Frank Shu presented his Trim Proposal to T13, which defines the ATA commands used by SATA interface. Exs. 1017-1018. Thus, POSITAs in the industry knew about Frank Shu's proposal to include the Trim command in the ATA standard used by SATA. Baker ¶ 176. Thus, when applying UTL's "SATA interface" interpretation, a POSITA would have found this element obvious. *Id.*

Although the Shu Patent does not detail the inner components of the SSD, a POSITA would have known that SSDs at the time generally included a controller, as referenced in IBM. Ex. 1027 at 536-37 (referencing the "SSF controller"). A POSITA would have recognized that this controller, together with the ATA interface, forms a circuit for receiving ATA requests and processing the requests by executing a software algorithm. Baker ¶¶ 177-178. Thus, when applying UTL's interpretation of "module," as "circuitry, software, and/or firmware," a POSITA would still have

found this element obvious. *Id.*

The Shu Patent teaches receiving a "remove-on-delete" command that comprises a logical identifier. Ex. 1003, 3:17-22 ("In one example, this includes sending file location information indicating the beginning of the file via logical block addressing ('LBA') typically followed by the length of the file to the SSD"); Ex. 1002, [0014]. Although this sentence of the Shu Patent does not explicitly mention the "remove-on-delete" or "Trim" command, to the extent the Shu Patent is unclear about what contains the LBA, a POSITA would have seen that Frank Shu clarified in his proposal that the Trim command contains an LBA. Ex. 1017 §§ 2-4 (describing same format using LBAs); Ex. 1018 § 6 (still including "LBA" in the format). Thus, a POSITA would have found it obvious that the message comprises a logical identifier. Baker ¶ 179.

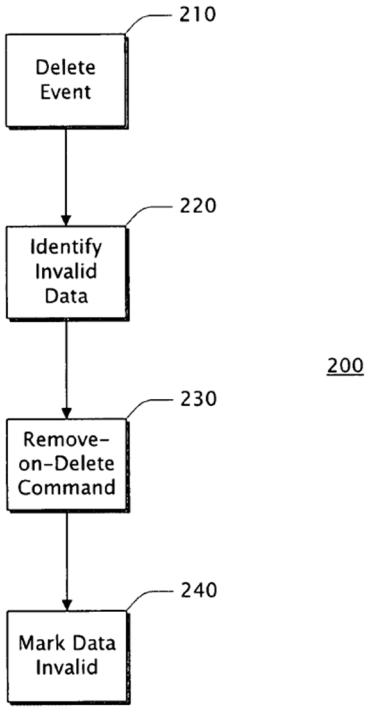
d) Element 1[d]

UTL accuses the remove-on-delete (Trim) command of infringing this element. Ex. 1013 at 2. But because the '658 Patent lacks priority to the 2006 Provisional as explained in Section VI above, the accused command anticipates this element as prior art under UTL's own theory, regardless of how UTL construes this element.

UTL contends, "The TRIM command tells the SSD that specific areas contain data …. From the user's perspective, this data has been deleted from a document."

Id.

The Shu Patent teaches that the SSD is configured to receive a "remove-on-delete" command in response to a delete event. Ex. 1003, abstract, 1:60-63, 3:20-22, 4:35-38, 4:65-5:4, Fig. 2 block 230 (reproduced below); Ex. 1002 at 23, [0013], [0014], [0020], [0023], Fig. 2.



An SSD receives this command when:

> A user, such as a person or system, may indicate via any suitable interface that some data, such as a file, should be deleted. The file system typically modifies a persistent data structure indicating the file has been deleted…. In a conventional computing system, the file system may be aware of files that are deleted, but an SSD is not.

Ex. 1003, 3:8-27; *see also id.*, 1:32-34, 1:55-63, 4:35-38, 5:3-11; Ex. 1002, [0014]; *see also id.*, [0001], [0003], [0020], [0023]-[0024]. Because the file system (e.g., as shown to a user via an operating system) is aware of what is deleted, this resembles the accused example: "From the user's perspective, this data has been deleted from a document." Ex. 1013 at 2. Alternatively, UTL may be accusing that the Trim command indicates "the file has been deleted." Ex. 1003, 3:12-13; Ex. 1002, [0014].

A POSITA would have recognized this "remove-on-delete" in the Shu Patent as the same "Trim" command proposed by Frank Shu to T13. Baker ¶¶ 156-157; Exs. 1017-1018. Both the "remove-on-delete" command and the Trim command have the same format (specifying an LBA), originated from the same person (Frank Shu), were published around the same time (in April 2007), and do the same thing (identify invalid data stored on an SSD). Ex. 1003, 3:17-20; Ex. 1002, [0014]; Ex. 1017 § 3; Ex. 1018 § 6.1.1.

Thus, when applying UTL's interpretation, this claim element would have been obvious to a POSITA. Baker ¶¶ 181-184.

e) Element 1[e]

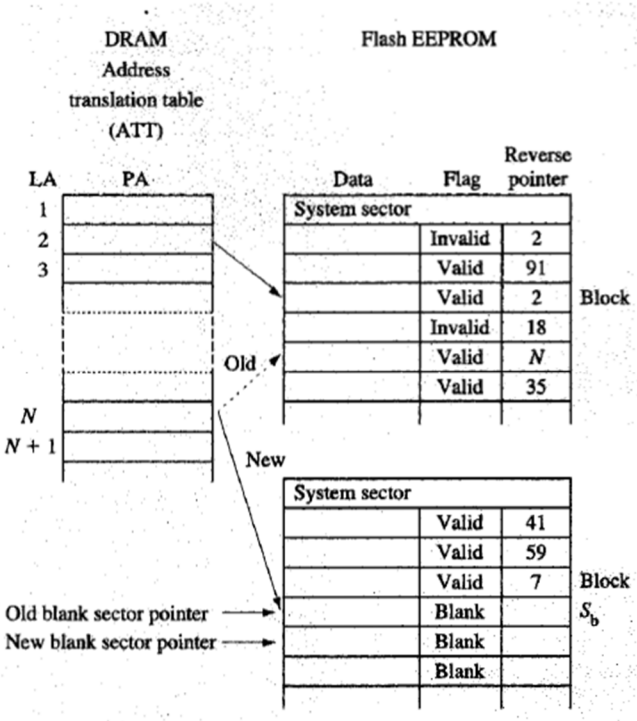
The Shu Patent discloses "mapping the data of the file being deleted to the corresponding data stored on an SSD." Ex. 1003, 4:51-54; Ex. 1002, [0022]. This

> mapping results in invalid data information identifying which data stored on the SSD are to be marked invalid…. The invalid data information identifies the exact data

> stored on the SSD that corresponds to the file being deleted and that is to be marked as invalid. Such data may be stored on the SSD as pages, blocks, or the like…. [A] remove-on-delete command … typically includes the invalid data information.

Ex. 1003, 4:52-66; Ex. 1002, [0022]-[0023]. To the extent the Shu Patent lacks an explicit recital of the mapping being a mapping of "logical identifier" and "physical storage location," a POSITA would have found this obvious because the Trim Proposal to SSD devices maps the LBAs to internal pages/blocks. Ex. 1017 at 3 ("A[] device will further remap LBA to its internal page and block for SSD."). Thus, a POSITA would have understood that the logical identifier in the Shu Patent's remote-on-delete/Trim command (*e.g.*, Ex. 1003, 3:17-22; Ex. 1002, [0014]; Ex. 1017 §§ 2-3) is mapped to physical storage locations in the form of blocks and/or pages of the SSD. Baker ¶ 185.

As further evidence of the widespread usage of logical-to-physical mapping tables, IBM figure 1 shows that such mapping tables were in use as far back as 1995.



Ex. 1027 at Fig. 1. "To allow flash EEPROMs to be used for SSFs,[6] we propose the *dynamic sector allocation* mechanism, in which logical sectors[7] are mapped dynamically to physical sectors on the SSFs in such a way as to avoid write-hot-spot sectors." *Id.* at 532. As part of this dynamic sector allocation (wear leveling), "[t]he

[6] "SSFs" stands for Solid-State Files, which refer to "secondary storage systems that use semiconductor memory devices instead of other storage media, such as magnetic storage devices." Ex. 1027 at 531. In other words, SSF refers to flash memory devices. Baker ¶ 186.

[7] A "sector" refers to the relevant physical unit of storage, such as a block or part of a block, such as a page. *Id.*; Ex. 1027 at Fig. 1 (showing sectors within a "block").

relation between the logical sector address and the physical sector address is stored in an *address translation table* (ATT) on volatile memory, such as SRAM or DRAM, in the SSF unit." *Id.* at 535.

Frank Shu envisioned sending the new remove-on-delete/Trim command to SSDs that have a logical-to-physical mapping table, such as the SSF of IBM.[8] Ex. 1003, 4:51-58; Ex. 1002, [0022]; Ex 1017 at 3 ("[D]evice will further remap LBA to its internal page and block for SSD."). Indeed, the remove-on-delete/Trim command informs the SSD about invalid "data associated with a delete event, and marking the deleted data stored on the SSD as invalid such that the SSD can avoid unnecessary operations on invalid data." Ex. 1003, 1:20-32, 1:55-59, 3:38-47, 4:29-31, 5:11-13; Ex. 1002, [0001], [0003], [0014], [0019], [0024]. These "unnecessary operations" refer to unnecessarily preserving invalid data during dynamic sector allocation (wear leveling) and garbage collection (which includes "merge" operations) as used in SSDs like IBM's SSF. Baker ¶ 187; Ex. 1027, *passim*; Ex. 1017 at 3 ("Such optimization is also applicable for SSD w[e]ar leveling

[8] IBM describes figure 1 for NOR flash EEPROM. Ex. 1027 at 535. Neither the Shu Patent nor the claims require NAND flash. Regardless, the same address mapping technique applies for NAND flash, with alternative grouping and bit representation adjustments for NAND explained in Exhibit 1027 at 537-38.

process."). These techniques use a logical-to-physical mapping table. Baker ¶¶ 163-165, 186-187; Ex. 1017 at 535-40. Thus, a POSITA would have found it obvious that the Shu Patent's SSD receives the remove-on-delete command with an LBA matching an entry in the SSD's address translation table, which maps the "relation between the logical sector address and the physical sector address" as described in IBM. Baker ¶ 187; Ex. 1027 at 535.

f) Element 1[f]

UTL interprets the "storage module" as "circuitry, software, and/or firmware" configured to store persistent data as claimed. Ex. 1013 at 3. According to UTL, the Trim command indicates that "the data associated with the logical identifier has been erased" because, "[f]rom the user's perspective, this data has been deleted from a document." *Id.* at 2. But regardless of how UTL interprets this claim, the accused Trim command predates the '658 Patent, which lacks priority to the 2006 Provisional as explained in Section VI above.

The Shu Patent discloses that a "user … may indicate via any suitable interface that some data, such as a file, should be deleted." Ex. 1003, 3:9-11; Ex. 1002, [0014]. Then, "[t]he file system typically modifies a persistent data structure indicating the file has been deleted, such as by removing a reference to the deleted file from a directory or the like." Ex. 1003, 3:12-15; Ex. 1002, [0014]. These actions result in the file data being deleted from "the user's perspective," as accused

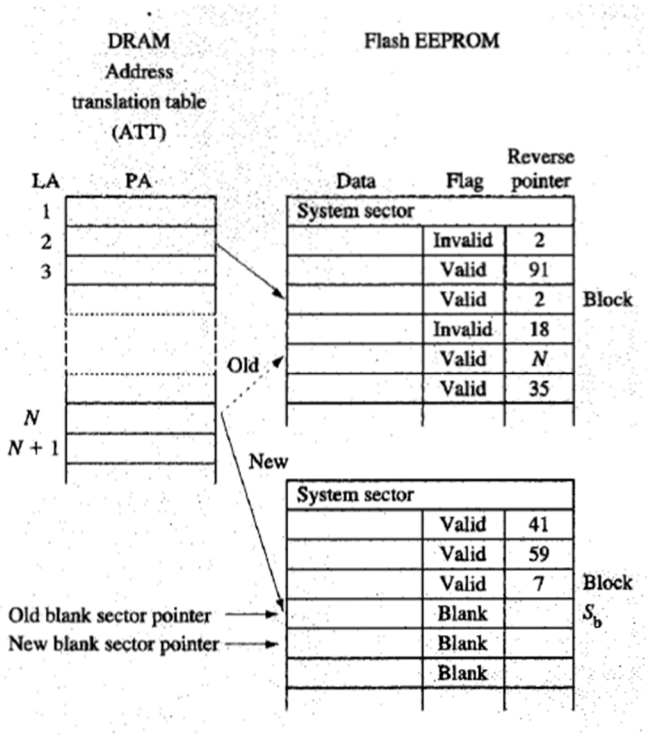
by UTL. Baker ¶ 190; Ex. 1013 at 2.

Then, the Shu Patent teaches to send the remove-on-delete/Trim command. Ex. 1003, 4:4-7, 4:26-34, 4:65-5:22, Figs. 1-3; Ex. 1002, [0017], [0019], [0023]. This remove-on-delete/Trim command "instructs the SSD device … to mark the indicated data as invalid…. Such a mark may take any form sufficient to identify the invalid data to the SSD device. Such marking may involve marking pages and/or blocks as invalid, depending on how the SSD structures its data." Ex. 1003, 4:67-5:9; Ex. 1002, [0023]-[0024]. As best understood, UTL alleges that this invalidity marking in the SSD is the claimed "persistent data" stored in response to the Trim command. Ex. 1013 at 2-3.

The Shu Patent does not explicitly state to store the invalidity marking as *persistent* data in the SSD, but a POSITA would have found this obvious because the purpose of an SSD is to store persistent data, not to lose the data after a transitory period, because flash memory is "nonvolatile." Baker ¶ 192. The references support this conclusion.

The Shu Patent teaches that the "marking may involve marking pages and/or blocks or the like as invalid, depending on how the SSD structures its data," and that "[s]uch a mark may take any form sufficient to identify the invalid data to the SSD device." Ex. 1003, 5:6-9; Ex. 1002, [0024]. A POSITA would have understood that a "sufficient" form refers to a persistent form, consistent with the SSD's purpose of

persistently storing data. Baker ¶¶ 192-193. As discussed in Section XI.A.e above, IBM provides an example of an SSF that Shu envisioned would receive the remove-on-delete/Trim command. The SSF in IBM stores the "Invalid" flag in the Flash EEPROM, a form of persistent memory:



Ex. 1027, Fig. 1. In view of the Shu Patent's instructions to mark data invalid, a POSITA would have found it obvious to set the corresponding "invalid" flag in the Flash EEPROM based on this IBM figure showing the invalid flags set in persistent flash memory. Baker ¶ 193. Moreover, a POSITA would have known to store all data, including the address translations and validity information, in persistent memory to maintain the data when computers are turned off; otherwise, the data would be lost when users turn off their computers. Ex. 1027 at 535 (discussing need

to preserve data through power-off to reconstruct the data in the DRAM cache); Baker ¶ 193. Thus, storing "persistent data on the non-volatile storage medium in response to the indication, wherein the persistent data is configured to indicate that the data associated with the logical identifier is erased," as interpreted by UTL, would have been obvious to a POSITA. Baker ¶ 193.

Moreover, a POSITA would have found it obvious that the SSD in the Shu Patent used a storage module, including circuitry that executes software, to perform the storage operations, like the "storage module" accused by UTL. *Id.* 194; Ex. 1027 at 535-36; Ex. 1013 at 3 (the "storage module" includes "circuitry, software, and/or firmware"). For example, the storage algorithms in IBM are implemented as software by a "controller" (circuitry) that executes software. Ex. 1027 at 535-36. Thus, when applying UTL's interpretation, this claim would have been obvious to a POSITA. Baker ¶ 194.

To the extent UTL argues that this claim requires further erasing the data in garbage collection and then changing the persistent "invalid" flag to "blank" or "erased" to indicate that the data was erased in garbage collection, even this would have been obvious to a POSITA. *Id.* 195.

Since 1995, IBM taught for flash devices to "[e]rase the selected cluster" (a

block[9]) in order to "automatically prepare[] blank sectors in the background," thereby increasing on-demand write speeds by avoiding the erase latency. Ex. 1027 at 532, 538-39. A blank status flag then "indicates that the sector has been erased." *Id.* at 536. Although valid data is copied (or "merged") into another location during garbage collection (*id.* at 539, step 3), any data indicated as invalid by the remove-on-delete/Trim would not be preserved during the merge operation, as the Shu Patent instructs. Ex. 1003, 4:29-34; Ex. 1002, [0019]. A POSITA would have found it obvious that the data indicated by the remove-on-delete/Trim command would be erased and then marked as "blank" in the flash EEPROM. Baker ¶ 196. Thus, a POSITA would also have known that the SSD's garbage collection algorithm executed by the controller (the storage module), in response to the remove-on-delete/Trim command, is configured to store a "blank" flag (the "persistent data") in the flash EEPROM (the "non-volatile storage medium") to indicate that the data associated with the logical identifier is erased during garbage collection. *Id*.

B. Claim 2

a) Element 2[a]

As discussed in Section XI.A.e (element 1[e]) above, the Shu Patent

[9] A *cluster* denotes "the unit that is erased at one time, which may consist of only one block." Ex. 1027 at 538.

envisioned sending the remove-on-delete/Trim command to an SSD (such as the SSF disclosed in IBM) that has mappings between logical identifiers and physical storage locations, like the ATT disclosed in IBM figure 1.

IBM further teaches that "the ATT is reconstructed during every SSF power-up sequence." Ex. 1027 at 535. "This approach requires information for reconstructing the ATT to be written on the EEPROM devices." *Id.* IBM discloses a "controller" for executing algorithms, so the controller software for reconstructing the ATT makes the claimed "index reconstruction module" obvious. *Id.* at 536; Ex. 1013 at 3 (a module is "circuitry, software, and/or firmware"); Baker ¶ 199. A POSITA would have found it obvious to reconstruct the ATT in RAM, which is faster than flash, for improved speeds based on the principles of memory caching. Thus, this claim element would have been obvious to a POSITA. Baker ¶¶ 199-200.

b) Element 2[b]

As discussed in Section XI.B.a (element 2[a]) above, Shu's SSD may be implemented like IBM's SSF, which reconstructs the address translation table. "This means that the information for reconstructing the ATT should always be maintained on the SSFs in preparation for power-off." Ex. 1027 at 535. Thus, in response to a remove-on-delete/Trim command, the data associated with the logical identifier in the trim command will be marked as "invalid" and then marked as "blank" after garbage collection. Baker ¶ 202; *see supra* Section XI.A.c (explaining

that remove-delete/Trim includes a logical identifier) and *supra* Section XI.A.f (explaining two ways of indicating that data of the logical identifier is erased).

The reconstructed ATT will indicate that data associated with the logical identifier (the logical address previously designated in the remove-on-delete/Trim command) is erased based on the persistent data in Flash EEPROM. Ex. 1027 at Fig. 1; Baker ¶ 203. In the reconstructed ATT table, the logical identifier is associated with a PA that points to a location in the persistent flash EEPROM flagging the data as either "invalid" (if before garbage collection completed, under UTL's theory that "invalid" indicates that data appears erased to a user) or "blank" (if garbage collection has completed, under the theory that "blank" means the data in flash memory is erased). Baker ¶ 203; Ex. 1027 at Fig. 1. Thus, the indication of erased data in the reconstructed ATT is based on the persistent data stored in the flash EEPROM. Baker ¶¶ 203-204; Ex. 1027 at Fig. 1. As discussed in Section XI.B.a (element 2[a]) above, the index reconstruction module includes IBM's controller that executes software algorithms. Thus, when applying UTL's interpretation, this claim element would have been obvious to a POSITA. Baker ¶¶ 203-204.

C. Claim 3

a) Element 3[a]

As discussed above in Section XI.A.e above, the SSD in the Shu Patent

includes "mappings" such as those shown in IBM's figure 1. POSITAs used "mapping" and "index" to mean the same thing. Baker ¶ 205. These mappings map "LAs" (logical addresses) to PAs (physical addresses), which are storage locations of the non-volatile storage medium. Ex. 1027 at Fig. 1. The mappings make up the index. Thus, a POSITA would have found this element obvious. Baker ¶ 205.

b) Element 3[b]

This claim element recites limitations indistinguishable from the limitations of claim 2[a]-[b] and would have been obvious to a POSITA for the same reasons. *Id.* ¶ 206.

D. Claim 4

a) Element 4[a]

This claim element recites limitations similar to the limitations of claim 3[a] except for requiring "a plurality of index entries." For the reasons discussed for claim 3[a], a POSITA would have understood this element obvious. *Id.* ¶ 207; Ex. 1027 at Fig. 1 (showing a plurality of entries in the ATT); *see supra* Section XI.C.a.

b) Element 4[b]

This claim element recites limitations indistinguishable from the limitations of claim 2[a]-[b] and would have been obvious to a POSITA for the same reasons. Baker ¶ 208; *see supra* Section XI.C.

E. Claim 5

As discussed in Section XI.D (claim 4) above, a POSITA would have found it obvious to send the remove-on-delete/Trim command to an SSD like the flash device of IBM and to use an index reconstruction module. IBM's garbage collection system will update the status indicator to "blank" to indicate that data is erased. Ex. 1027 at 532 ("*background garbage collection* mechanism ... automatically prepares blank sectors in the background, so that, under ideal conditions, the erasure time can be completely hidden."), 536 ("Blank indicates that the sector has been erased."), Fig. 1 (showing updating the index to show "Blank" status).

The flash device may be NOR- or NAND-based flash. Ex. 1027 at 533. IBM teaches minor differences (different flags values, optional NAND block clustering, etc.) in the implementations between NAND and NOR, but the logical to physical mapping of the ATT shown in Ex. 1027 figure 1 applies to both. Baker ¶ 211. NAND and NOR flash devices were the two limited, obvious, and popular choices of flash drives known to POSITAs, as evidenced by IBM addressing both implementations; selecting one over the other was simply a design choice for a specific application. *Id.* In NAND implementations, the ATT also includes the invalid status flag. Ex. 1027 at 541 ("When the FAT is modified by an erase command, the FAT monitor instructs the SSF to mark the appropriate data sectors invalid, since they are logically erased. (For NAND architectures, the FAT monitor

marks the deleted sectors *invalid* in the ATT entry...)").

Thus, a POSITA would have understood that the ATT update occurs when the remove-on-delete/Trim command is sent to the flash memory device to indicate invalid data (a logical erasure). Baker ¶ 212. In response, the ATT (the claimed "index") is updated with a mark that indicates that the deleted sectors are invalid. This indication in the ATT will update to "blank" after subsequent garbage collection, as explained in Sections XI.B.b and XI.A.f above. Thus, a POSITA would have found this claim element obvious in view of the Shu Patent and IBM's SSF. *Id.*

F. Claim 8

UTL accuses an SSD of infringing this element for responding to read requests by returning an indication that data "is cleared to zero as such an indication." Ex. 1013 at 4. This functionality was specified in Frank Shu's Trim Proposals, revision 1, which predate the priority date of the asserted claims as explained in Sections VI and XI.B above. Ex. 1018.

The Shu Trim Proposal, revision 1, has a section titled "Deallocate(Trim)" that describes how to respond to read requests of deallocated/trimmed data: "If a read occurs to any part of the data set before it is written, the device . . . may return all 0s." *Id.* at 7. The condition "before it is written" means that even though the original data has been logically erased or marked as invalid, the data still remains in

the physical storage location because the data has not yet been rewritten as all one's or all zero's (e.g., during physical erase or garbage collection). Baker ¶ 215. A POSITA would have considered the SSD's controller that runs software to implement this function to be the "read request module." *Id*. Thus, when applying UTL's interpretation, this claim would have been obvious to a POSITA because Frank Shu's Trim Proposals specified it for the ATA industry standard. *Id*.

G. Claim 9

UTL appears to accuse an invalid-type mark in response to the Trim command as infringing this element. Ex. 1013 at 2, 4.

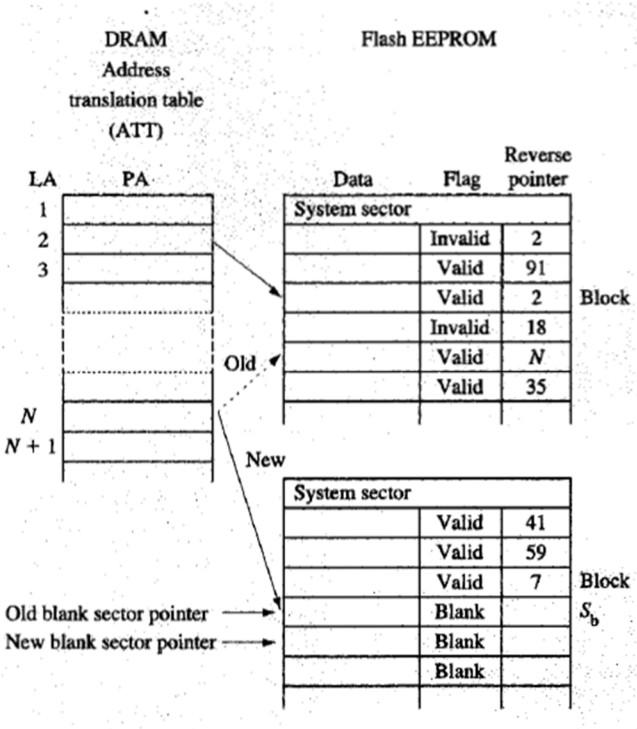
As discussed in Section XI.A.f (claim 1[f]) above, the remove-on-delete/Trim command "instructs the SSD device … to mark the indicated data as invalid…. Such a mark may take any form sufficient to identify the invalid data to the SSD device. Such marking may involve marking pages and/or blocks as invalid, depending on how the SSD structures its data." Ex. 1003, 4:67-5:9; Ex. 1002, [0023]-[0024]. Thus, a POSITA would have found it obvious for the SSD to record that the data associated with the LBA in the remove-on-delete/Trim command is "invalid," as taught by IBM. Baker ¶ 219; Ex. 1027 at Fig. 1 (showing "invalid" recorded for an LA and PA pair). "Invalid indicates that the sector holds old data (garbage)" that will be erased during garbage collection. Ex. 1027 at 536, 539. "This frees the SSD device from performing any operations to preserve or maintain such data." Ex. 1003,

4:39-41, 5:11-13; Ex. 1002, [0020], [0024].

A POSITA would have found the claimed "marking module" obvious because IBM discloses a controller (a circuit) to run a software "control algorithm" for marking data invalid. Baker ¶ 220; Ex. 1027 at 535, 537-39; Ex. 1013 at 2 (generally construing "module" as "circuitry, software, and/or firmware"). Thus, this claim, as interpreted by UTL, would have been obvious to a POSITA. Baker ¶¶ 219-221.

H. Claim 10

As discussed in Section XI.G (claim 9) above, a POSITA would have understood that the Shu Patent's SSD would have a "marking module" that records that the data associated with the LBA in the remove-on-delete/Trim command is "invalid," as taught in IBM. *Id.* ¶ 223; Ex. 1027 at Fig. 1 (showing "invalid" recorded, reproduced below).



As illustrated, the flash EEPROM includes index entries. Ex. 1027 at Fig. 1. Each index entry has a physical address and includes fields for the user data, status flag, and reverse pointer. *Id.* at 536, Fig. 1. Of these, the reverse pointer is the logical address. Thus, these index entries associate a logical address with a physical address. Baker ¶ 224. It would have been obvious to a POSITA that the marking module would flag the index entry as invalid in response to a remove-on-delete/Trim command that includes the logical address mapped to that index entry. *Id.* ¶¶ 222-225.

I. Claim 11

According to UTL, this element is met by "circuitry and/or software/firmware" that implements "garbage collection" to "return[] the physical

memory to the point where it can be written again." Ex. 1013 at 5. A POSITA would have understood that the Shu Patent's SSD would also use garbage collection in the same way, as evidenced by IBM. Baker ¶¶ 227, 229. Indeed, the inventors admit that "[g]arbage collection" was "commonly used." Ex. 1015 at 40.

When applying UTL's interpretation, a POSITA would have found the claimed "storage recovery module" obvious because IBM discloses a controller (a circuit) to run software to manage erasure and execute the garbage collection algorithm. Baker ¶ 228; Ex. 1027 at 537-39.

The Shu Patent specifically described sending the remove-on-delete/Trim command to SSDs that used garbage collection. The Shu Patent sought to inform the SSD about invalid "data associated with a delete event, and marking the deleted data stored on the SSD as invalid such that the SSD can avoid unnecessary operations on invalid data," such as "merge" operations. Ex. 1003, 1:20-25, 1:55-59, 3:38-40, 4:29-31, 5:11-13; Ex. 1002, [0001], [0003], [0014], [0019], [0024]. These merge operations occur during garbage collection. Ex. 1027 at 539, step 3.

Since 1995, IBM taught for flash devices to use "[b]ackground garbage collection," in order to "automatically prepare[] blank sectors in the background," thereby increasing on-demand write speeds by avoiding the erase latency. *Id.* at 532, 538-39. As part of garbage collection, valid data is copied (or "merged") into another location (*id.* at 539, step 3), but any data indicated as invalid by the remove-

on-delete/Trim command would not be preserved during the merge operation, as the Shu Patent instructs. Ex. 1003, 4:29-34; Ex. 1002, [0019]. Then, the selected cluster/block[10] is erased. Ex. 1027 at 539, step 4. Erasing the block returns, or recovers, the block to a state where it can be written to again. *Id.* at 533. Thus, the block would be erased/recovered in response to the remove-on-delete command, rather than merged and preserved, such as how UTL interprets the claim. Baker ¶ 230.

Thus, a POSITA would have found it obvious for the SSD in the Shu Patent to use garbage collection, as taught by IBM, because garbage collection was commonly used for improving on-demand write speeds and because the Shu Patent specifically taught to send the remove-on-delete/Trim command to SSDs that used merge operations, such as in garbage collection. *Id.* ¶ 231. When applying the UTL's interpretation, a POSITA would have found the claim obvious. *Id.*, ¶¶ 227-231.

J. Claim 12

According to UTL, this element is met by "garbage collection." Ex. 1013 at 5. As discussed for claim 11 in Section XI.I (claim 11) above, a POSITA would

[10] A *cluster* denotes "the unit that is erased at one time, which may consist of only one block." Ex. 1027 at 538.

have found it obvious for the SSD in the Shu Patent to use garbage collection, which includes the step, "Erase the selected cluster," referring to erasing a block. Ex. 1027 at 538 (defining cluster), 539 at step 4. This block, indicated as invalid by the remove-on-delete/Trim command, would not be copied during the preceding step (*id.* at 539, step 3), as the Shu Patent instructs. Ex. 1003, 4:29-34; Ex. 1002, [0019]; Baker ¶ 223. Thus, by applying UTL's interpretation, this block will be erased in response to the remove-on-delete/Trim command. Baker ¶ 223.

A POSITA would have found the claimed "erase module" obvious because IBM discloses a controller (a circuit) to run software to manage erasure and execute the garbage collection algorithm. *Id.* ¶ 234; Ex. 1027 at 537-39.

XII. Secondary Considerations

Simultaneous invention by others shows that the claims fall within the level of the ordinary skill in the art. "Independently made, simultaneous inventions, made within a comparatively short space of time, are persuasive evidence that the claimed apparatus was the product only of ordinary mechanical or engineering skill." *Geo. M. Martin Co. v. All. Mach. Sys. Int'l LLC*, 618 F.3d 1294, 1305 (Fed. Cir. 2010). The Board has held that exhibits of a standard-setting group on a related standard "are evidence of simultaneous invention by others," support finding challenged claims obvious, and "are persuasive evidence that the claimed apparatus 'was the product only of ordinary mechanical or engineering skill.'" *ZTE (USA) Inc. v.*

Evolved Wireless LLC, No. IPR2016-00757, Paper 42 at 29 (P.T.A.B. Nov. 30, 2017).

Here, Exhibits 1017-1018 show that standard-setting group T13 began work on the Trim command proposal at least by April 21, 2007, months before September 2007. Baker ¶ 236. UTL accuses this Trim command of infringing the claims. Ex. **1013, passim. Like the ZTE case, here a standard-setting group worked on the same** technology around the same time. Exs. 1017-1018. Also, other prior art taught similar commands. Ex. 1029, 17:52-56 (an erase command "specifies the (logical) sectors to be erased"); Ex. 1028, 9:2-3 ("logical block address ... designated in the erase command"). Furthermore, many claim elements were already well known in the art. *See, e.g.*, Ex. 1020 §§ 2.2 (Ban patented the FTL in 1995, and the FTL became part of an industry standard), 2.3 (explaining the garbage collection process); Ex. 1027 at 538-39 (explaining the garbage collection process). Thus, Exhibits 1002-1003 and 1017-1018 all serve as evidence of simultaneous invention by others, and the Board should find the challenged claims obvious for being only the product of ordinary mechanical or engineering skill.

XIII. Mandatory Notices

A. Real Parties-in-Interest

The named Petitioners are the only entities who are funding and controlling this Petition and are therefore all named as real parties-in-interest. No other entity

is funding, controlling, or otherwise has an opportunity to control or direct this Petition or Petitioner's participation in any resulting IPR.

Out of an abundance of caution, Petitioners also identify Dell Technologies Inc., Dell Inc., Denali Intermediate Inc. (which is a corporate parent entity of Dell Inc.), and HP Inc. as a real parties-in-interest. UTL sued Dell Technologies Inc., Dell Inc., and HP Inc., alleging infringement of the challenged patent, but those cases were dismissed before the filing of this Petition.

Petitioners also identify that there are many entities such as suppliers, resellers, part providers, contractors, etc., who may have financial liabilities with respect to the hundreds of accused products in the related litigations. Petitioners do not believe that any of these entities, however, are real parties-in-interest. None of these other entities participated in the preparation or funding of this Petition or otherwise had an opportunity to control or direct this Petition. To Petitioners' best knowledge, no entity, other than Petitioners and the entities named in this Section XIII, has been served with a complaint alleging infringement of the patent at issue herein.

B. Related Proceedings

UTL asserted the '658 Patent against Petitioners in the Western District of Texas, Case No. 6:20-cv-500. In the same court, UTL also asserted the '658 Patent against Dell Technologies Inc. and Dell Inc. in Case No. 6:20-cv-499 and against

HP Inc. in Case No. 6:20-cv-501. UTL filed each lawsuit on June 5, 2020.

Three IPR proceedings relate to the same patent family: IPR2021-00343 (Pat. No. 8,533,406), IPR2021-00344 (Pat. No. 8,762,658), and IPR2021-00345 (Pat. No. 9,632,727). Petitioners are also filing contemporaneously two additional IPR proceedings that challenge the priority date of the same patent family: IPR2021-00940 (Pat. No. 8,533,406) and IPR2021-00942 (Pat. No. 9,632,727).

C. Lead and Backup Counsel

The following lead and backup counsel represent Petitioners:

Lead Counsel for Petitioner	Backup Counsel for Petitioner
Katherine A. Vidal Winston & Strawn LLP 275 Middlefield Rd., Suite 205 Menlo Park, CA 94025 kvidal@winston.com T: 650.858.6425, F: 650.858.6550 USPTO Reg. No. 46,333	Michael Rueckheim Winston & Strawn LLP 275 Middlefield Rd., Suite 205 Menlo Park, CA 94025 mrueckheim@winston.com T: 650.858.6433, F: 650.858.6550 (to seek *pro hac vice* admission) Qi (Peter) Tong Winston & Strawn LLP 2121 N Pearl St. Dallas, TX 75201 ptong@winston.com T: 214.453.6473, F: 214.453.6400 USPTO Reg. No. 74,292

D. Electronic Service

Petitioners consent to electronic service at:

Winston-IPR-Unification@winston.com

XIV. Fees

Petitioners have paid the required fee electronically through P.T.A.B. E2E.

XV. Conclusion

Petitioners respectfully request that the Board institute IPR and enter a final written decision finding the challenged claims unpatentable.

Dated: June 4, 2021

Respectfully submitted,

/s/ *Katherine A. Vidal*

Katherine A. Vidal
Winston & Strawn LLP
275 Middlefield Rd, Suite 205
Menlo Park, California 94025
kvidal@winston.com
T: 650.858.6425, F: 650.858.6550
USPTO Reg. No. 46,333
Lead Counsel for Petitioners Micron Technology, Inc.; Micron Semiconductor Products, Inc.; and Micron Technology Texas LLC

Michael Rueckheim
Winston & Strawn LLP
275 Middlefield Rd, Suite 205
Menlo Park, California 94025
mrueckheim@winston.com
T: 650.858.6433, F: 650.858.6550
Backup Counsel for Petitioners Micron Technology, Inc.; Micron Semiconductor Products, Inc.; and Micron Technology Texas LLC
(to seek *pro hac vice* admission)

Qi (Peter) Tong
Winston & Strawn LLP
2121 N Pearl St,
Dallas, TX 75201
ptong@winston.com
T: 214.453.6473, F: 214.453.6400
USPTO Reg. No. 74,292
Backup Counsel for Petitioners Micron Technology, Inc.; Micron Semiconductor Products, Inc.; Micron Technology Texas LLC

CLAIM LISTING

Claim 1	
Element	**Language**
1[a]	An apparatus for managing data stored on a non-volatile storage medium, comprising:
1[b]	a non-volatile storage medium;
1[c]	a request receiver module configured to receive a message comprising a logical identifier,
1[d]	the message indicating that data associated with the logical identifier has been erased,
1[e]	wherein the logical identifier is mapped to a physical storage location of the non-volatile storage medium; and
1[f]	a storage module configured to store persistent data on the non-volatile storage medium in response to the indication, wherein the persistent data is configured to indicate that the data associated with the logical identifier is erased.

Claim 2	
Element	**Language**
2[a]	The apparatus of claim 1, further comprising an index reconstruction module configured to reconstruct mappings between logical identifiers and physical storage locations of the non-volatile storage medium from contents of the non-volatile storage medium,
2[b]	wherein the reconstruction module is configured [to] indicate that data of the logical identifier are erased based on the persistent data stored on the non-volatile storage medium.

Claim 3	
Element	**Language**
3[a]	The apparatus of claim 1, further comprising: an index comprising mappings between logical identifiers and physical storage locations of the non-volatile storage medium; and
3[b]	an index reconstruction module configured to reconstruct the mappings using contents of the non-volatile storage medium, wherein the index reconstruction module is configured to indicate that data associated with the logical identifier are erased based on the persistent data stored on the non-volatile storage medium.

Claim 4	
Element	**Language**
4[a]	The apparatus of claim 1, further comprising: an index comprising a plurality of index entries comprising mappings between logical identifiers and physical storage locations of the non-volatile storage medium; and
4[b]	an index reconstruction module configured to reconstruct the index entries from data stored on the non-volatile storage medium, wherein the index reconstruction module is configured to record an indication that data associated with the logical identifier are erased based on the persistent data stored on the non-volatile storage medium.

Claim 5	
Element	**Language**
5	The apparatus of claim 4, wherein the index reconstruction module is configured to update the index to indicate that the data associated with the logical identifier are erased.

Claim 8	
Element	**Language**
8	The apparatus of claim 1, further comprising a read request module configured to return an indication that the data of the logical identifier are erased in response [to] a request pertaining to the logical identifier received while the data associated with the logical identifier remains on the physical storage location of the non-volatile storage medium.

Claim 9	
Element	**Language**
9	The apparatus of claim 1, further comprising a marking module configured to record that contents of the physical storage location associated with the logical identifier no longer need to be retained on a non-volatile storage medium in response to the indication.

Claim 10	
Element	**Language**
10	The apparatus of claim 1, further comprising a marking module configured to invalidate an index entry configured to associate the logical identifier with the physical storage location in response to the indication.

Claim 11	
Element	**Language**
11	The apparatus of claim 1, further comprising a storage recovery module configured to recover a storage division comprising the physical storage location in response to the indication.

Claim 12	
Element	**Language**
12	The apparatus of claim 1, further comprising an erase module configured to erase the physical storage location in response to the indication.

CERTIFICATE OF COMPLIANCE

This Petition complies with the word count limits set forth in 37 C.F.R. § 42.24(a)(1)(i) because this Petition contains 13,196 words, excluding the parts of the Petition exempted by 37 C.F.R. § 42.24(a)(1) and determined using the word count provided by Microsoft Word, which counsel used to prepare this Petition.

Dated: June 4, 2021

Respectfully submitted,
/s/ *Katherine A. Vidal*
Katherine A. Vidal
kvidal@winston.com
USPTO Reg. No. 46,333
Winston & Strawn LLP
275 Middlefield Rd, Suite 205
Menlo Park, California 94025
T: 650.858.6425, F: 650.858.6550

CERTIFICATE OF SERVICE

Under 37 C.F.R. §§ 42.6(e) and 42.105(a), this is to certify that on June 4, 2021, I caused to be served a true and correct copy of the above "**PETITION FOR *INTER PARTES* REVIEW OF CLAIMS 1-5 AND 8-12 OF U.S. PATENT NO. 8,762,658**" and Exhibits 1001-1040 by FedEx on UTL at the correspondence address of record for U.S. Patent No. 8,762,658:

Western Digital, c/o Longitude Licensing Stoel Rives LLP
201 South Main Street, Suite 1100
One Utah Center
Salt Lake City UT 84111

A courtesy copy of this Petition and supporting material was also served on litigation counsel for UTL via email:

Ed Nelson
Nelson Bumgardner Albritton PC
3131 W. 7th Street, Suite 300
Fort Worth, TX 76107
Email: ed@nbafirm.com

Dated: June 4, 2021

Respectfully submitted,

/s/ *Katherine A. Vidal*
Katherine A. Vidal
kvidal@winston.com
USPTO Reg. No. 46,333
Winston & Strawn LLP
275 Middlefield Rd, Suite 205
Menlo Park, California 94025
T: 650.858. 6425, F: 650.858.6550