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. IPR2021-00942

U.S. Patent No. 9,632,727

PETITION FOR *INTER PARTES* REVIEW OF CLAIMS 1-6 AND 12-16 OF U.S. PATENT NO. 9,632,727

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

Cases

Statutes

Other Authorities

PETITIONERS' EXHIBIT LIST

Ex. No.	Brief Description
1001	U.S. Pat. No. 9,732,727 B2, titled "SYSTEMS AND METHODS FOR IDENTIFYING STORAGE RESOURCES THAT ARE NOT IN USE," to Flynn et al. ("'727 Patent").
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. 9,732,727 (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. 8,762,658 B2, titled "SYSTEMS AND METHODS FOR PERSISTENT DEALLOCATION," 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 C to Plaintiff's First Amended Infringement Contentions: Unification Technologies' Allegations of Infringement with Respect to U.S. Patent No. 9,632,727, *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. 9,632,727, titled "SYSTEMS AND METHODS FOR IDENTIFYING STORAGE RESOURCES THAT ARE NOT IN USE," 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. 5,404,485, 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 comparison of the Shu Provisional Disclosure with 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	IDS submitted on September 25, 2013 during prosecution of U.S. Pat. No. 8,762,658.
1039	Declaration of Frank Shu Regarding Publication of Proposals (Exs. 1018, 1019), 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 U.S. Patent No. 9,632,727 (the "'727 Patent") should never have issued. In 2012, the Advance Technology ("ATA") standard adopted the "Trim" command that Frank Shu first proposed in April 2007. Ex. 1005; Ex. 1017. In 2014, the applicant wrote new claims that allegedly encompass the Trim command, but the claimed scope lacks support in the earliest provisional application. When applying the proper priority date—the September 22, 2007 filing date[1] of the second provisional application ("2007 Provisional")—the allegedly infringing Trim command becomes invalidating prior art.

II. Petitioners Meet Standing and Eligibility Requirements for *Inter Partes* Review

Petitioners certify under 37 C.F.R. § 42.104(a) that the '727 Patent "is available for *inter partes* review and that the petitioner is not barred or estopped from requesting an *inter partes* review challenging the patent claims on the grounds identified in the petition." Unification Technologies LLC ("UTL") sued Petitioners for alleged infringement less than one year ago on June 5, 2020. Exs. 1012, 1016.

[1] Petitioners assume the 2007 Provisional application supports the challenged claims because the prior art herein predate September 22, 2007. Petitioners reserve the right to challenge this assumption in the co-pending district court litigation.

III. Prosecution History of the ’727 Patent

The applicant filed the application for the ’727 Patent in 2014, two years after the Trim command became part of the ATA standard. Ex. 1021; Ex. 1005. During examination, the Examiner issued two rejections. Ex. 1021 at 273-84, 331-37. In response to the nonfinal rejection, the applicant made various amendments, including specifying that the indexer is “comprised within the solid-state storage controller” to overcome Patent Publication No. 2007/0136555 to Sinclair. *Id.* at 316-19, 352-56. Sinclair taught a “FAT table, which is stored by the host,” which the applicant distinguished by arguing that the claims required the solid-state storage controller, not the host operating system/host CPU, to maintain the indexer/table. *Id.* at 321.

During prosecution of a parent application that ultimately issued as Patent No. 8,762,658 (the “’658 Patent”), the applicant disclosed a December 12, 2007 proposal by Frank Shu to the T13 standard in an information disclosure statement (“2013 IDS”). Ex. 1011 at 5 (listing reference); Ex. 1038 at 13 (cite no. D19). The 2013 IDS failed to indicate that the proposal was the “sixth revision” of Frank Shu’s proposal or suggest the possibility of earlier revisions that predate the 2007 Provisional. Ex. 1014, cover. Because the 2007 Provisional predated the cited December 12, 2007 date, the Examiner likely signed the 2013 IDS without considering the sixth revision and without realizing that earlier proposals existed.

The Examiner allowed the application for the ’727 Patent.

IV. Technology Background

Flash memory is a form of solid-state nonvolatile computer memory organized in erasable units called “blocks,” which are made up of smaller “pages.” Ex. 1004 (Expert Declaration of Jacob Baker, hereinafter “Baker”) ¶ 122.

Since 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 have 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.

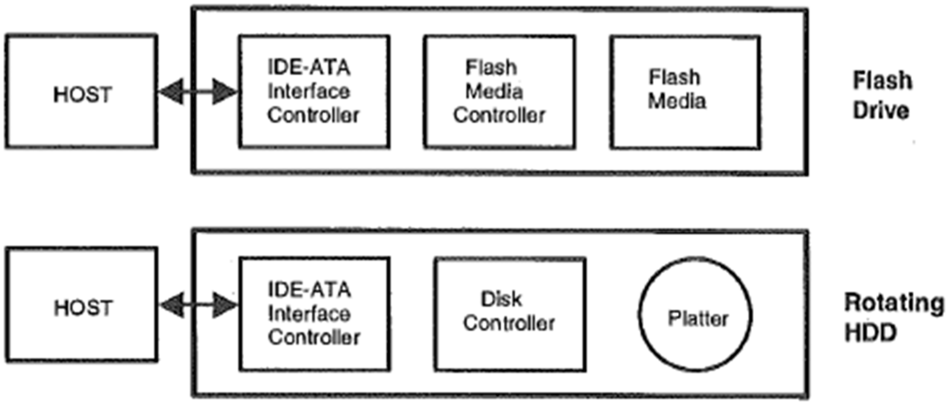


Figure 4.14: Mass Storage Architecture

Flash memory has long used a flash translation layer (“FTL”) to map logical addresses to physical addresses, and the industry widely credits Ban as inventing the FTL in 1995. Baker ¶ 143; Ex. 1027 Fig. 1 (showing FTL translating addresses); *see infra* Section X.C. 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 ¶ 144.

Unlike magnetic-platter hard drives, flash memory cannot be directly overwritten—a block must be erased before written to again. *Id.* ¶ 133. 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. 1001, 1017. Frank Shu then led Microsoft to announce that Windows would support the Trim command. Ex. 1022 at 8; Ex. 1023; Ex. 1024 at 2, 4, 10. The industry then adopted the Trim command as part of the ATA standard. Ex. 1005 § 7.9.3.2.

V. Summary of the '727 Patent

Independent claim 1 recites an apparatus including a solid-state storage medium, a solid-state storage controller, and an indexer. Ex. 1001, 53:22-40. The controller is "configured to implement storage operations." *Id.* The indexer is "configured to assign logical addresses … to physical addresses." *Id.* The indexer is also "configured to remove an assignment between an identified logical address and a physical address … in response to a message received from a host operating system, the message indicating that the identified logical address is erased," *Id.* UTL interprets this disclosure to mean indicating that data identified by the logical address is erased. *See infra* Section VIII (construing claims).

Independent claim 12 recites a system comprising a storage interface, a storage processor, a flash memory device, and a logical-to-physical translation layer. Ex. 1001, 54:42-64. The logical-to-physical translation layer "maps logical block addresses to corresponding respective physical block addresses." *Id.* The processor is configured to "receive … an empty-block directive command and a range of logical block addresses, update the logical-to-physical translation layer to indicate that data stored in the physical block addresses … do not need to be preserved, and store persistent data … indicating that the data … is deleted at the storage client." *Id.*

UTL contends that the "message" in claim 1 and the "command" in claim 12 cover Frank Shu's Trim command. Ex. 1013, *passim*.

VI. The '727 Patent's Priority Date Cannot Precede September 22, 2007

The disclosure of the '727 Patent diverged greatly from the 2006 Provisional. Indeed, inventor Jonathan Thatcher refused to sign the inventor declaration for the '727 Patent. Ex. 1021 at 236-37.

The 2006 Provisional cannot support the priority date for the challenged claims because it lacks support for the independent claim elements challenged in Sections (C)-(G) below. The 2006 Provisional also fails to support the dependent claims. UTL contends that parts of the 2006 Provisional provide support (Ex. 1019 at 39-40), 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). "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. Faulding Inc.*, 230 F.3d 1320, 1323 (Fed. Cir. 2000).

[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.

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) (citing cases).

"Inherency … 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).

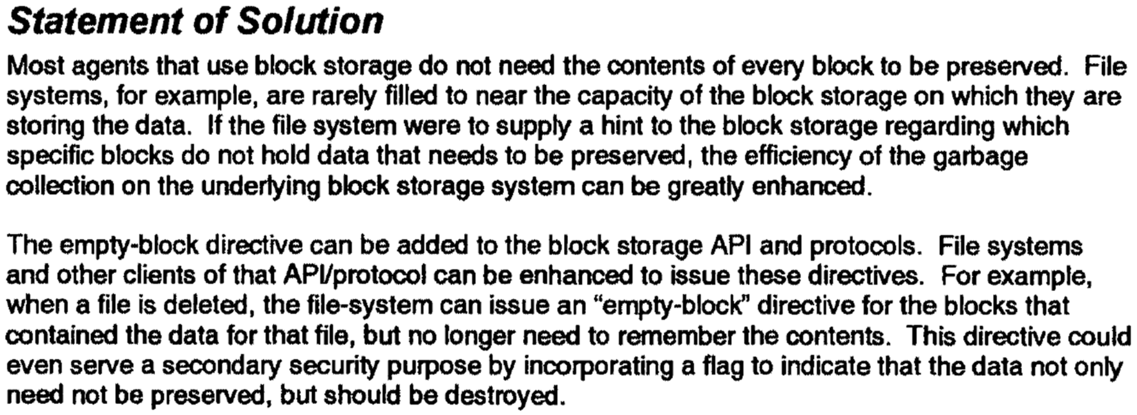
When filing an amendment to pending claims, "an applicant should show support in the original disclosure for new or amended claims." MPEP § 2163(II)(A)(3)(b); *see also* MPEP §§ 714.02, 2163.06. The applicant never argued that the 2006 Provisional supports the challenged claim amendments. Ex. 1021 at 316-25.

B. Summary of the 2006 Provisional

The 2006 Provisional contains a scattershot of separate ideas catalogued as "claims," none of which match the '727 Patent 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, 10-12 (describing "identical card pairings").

UTL contends that provisional "claim" 18 supports '727 Patent claims. Ex. 1030 at 10-11 (highlighting the "empty block" directive of provisional "claim" 18).

"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":

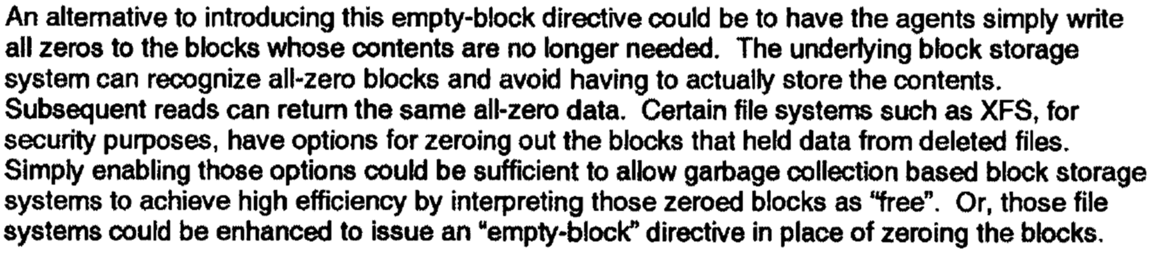


Statement of Solution

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. No Support for an Index/Logical-to-Physical Translation Layer in a Storage Controller/Processor in Claims 1 and 12

'727 Patent claim 1 requires "an indexer, comprised within the solid-state storage controller" configured to "implement storage operations on the solid-state storage medium in response to requests from a computer system." Ex. 1001. Claim 12 requires "a logical-to-physical translation layer maintained by the storage processor." *Id.* The context of both claims 1 and 12 requires the "storage controller" or "storage processor" to be part of the storage system, not the CPU of a host or client computer. Ex. 1021 at 321-22 (distinguishing tables maintained by hosts during prosecution).

The 2006 Provisional arguably discloses a block index and mapping in provisional "claims" 22 and 24. *Id.* at 61. However, provisional "claims" 22 and 24 make no mention of the empty-block hint/directive and says nothing about whether who or what is controlling the block index or mapping. *Id.* The 2006 Provisional leaves a POSITA to wonder whether the operating system CPU or the storage device's controller/processor should manage the mapping. Baker ¶ 72. The applicant emphasized this distinction in an amendment to overcome a rejection. Ex. 1021 at 316 (amending claim 31, issued as claim 1). By narrowing its claims based on features not supported by the 2006 Provisional, the claim lost priority. MPEP § 2163.05(II) (citing cases).

The patent owner cannot turn to provisional "claim" 27, titled "NAND Controller," for support. Ex. 1015 at 85. This section discusses using the NAND Controller for other tasks. *Id.* at 74-90 (listing other functions A-K). To the extent provisional "claim" 27 contains a passing reference to a controller getting a "map" from elsewhere, this means that the map is maintained elsewhere and, in any event, fails to describe the storage controller being configured to "assign logical addresses … to physical addresses," "remove an assignment," or "update the logical-to-physical translation layer to indicate that data … do not need to be preserved" as recited in challenged claims 1 and 12. Baker ¶¶ 73-74; Ex. 1001, 53:30-40, 54:45-60.

The applicant arguably (not admittedly) added support on Dec. 6, ***2007*** in related application 11/952,113 ("the '113 App."), which issued as Patent 8,261,005. Ex. 1034; Ex. 1001, cover. This '113 App. shows an object "index module" in a "Storage Controller" inside the "Solid-State Storage Device," rather than in processors of client computers. Ex. 1034, Figs. 1A, 1C, 2A (1A and 2A annotated and reproduced below). The 2006 Provisional contains no similar disclosure, and this absence supports the lack of priority. Baker ¶ 77.

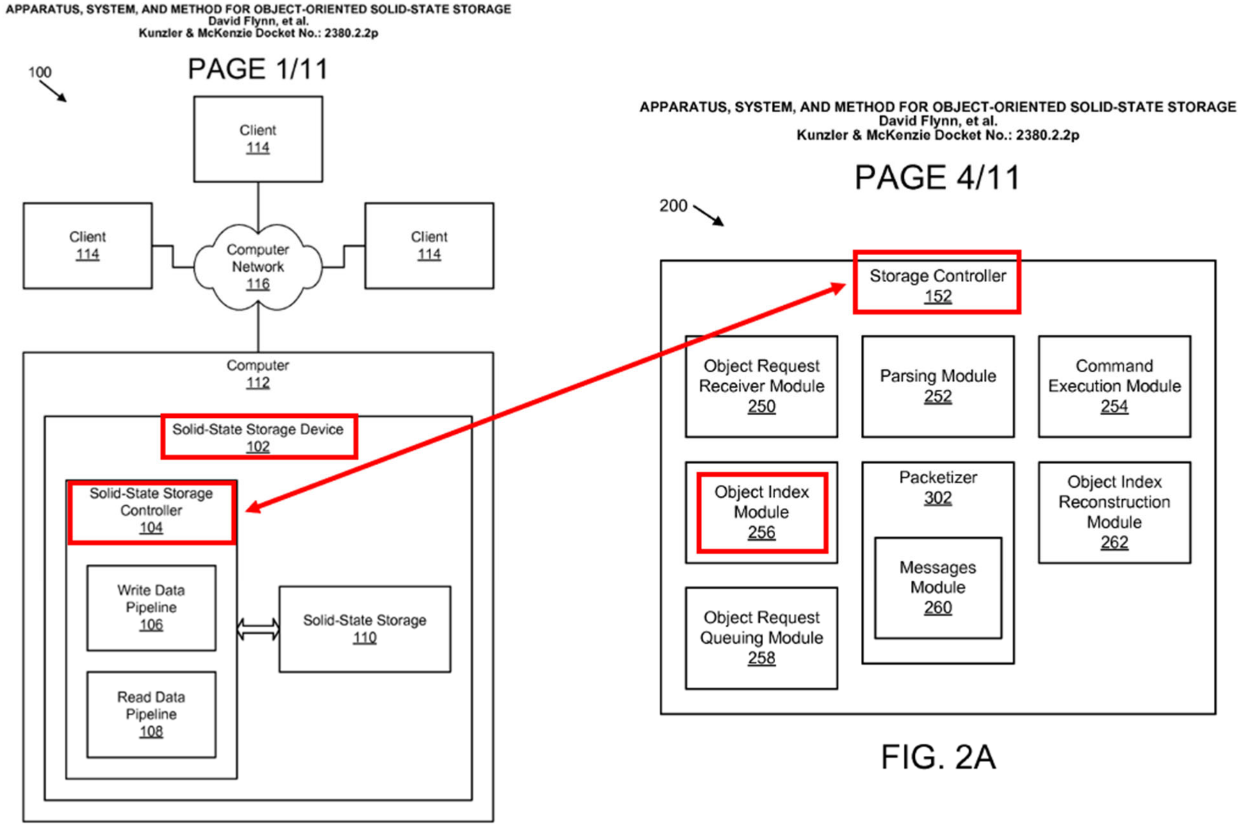

APPARATUS, SYSTEM, AND METHOD FOR OBJECT-ORIENTED SOLID-STATE STORAGE
David Flynn, et al.
Kunzler & McKenzie Docket No.: 2380.2.2p
PAGE 1/11
100
Client 114
Client 114
Computer Network 116
Client 114
Computer 112
Solid-State Storage Device 102
Solid-State Storage Controller 104
Write Data Pipeline 106
Solid-State Storage 110
Read Data Pipeline 108
APPARATUS, SYSTEM, AND METHOD FOR OBJECT-ORIENTED SOLID-STATE STORAGE
David Flynn, et al.
Kunzler & McKenzie Docket No.: 2380.2.2p
PAGE 4/11
200
Storage Controller 152
Object Request Receiver Module 250
Parsing Module 252
Command Execution Module 254
Object Index Module 256
Packetizer 302
Object Index Reconstruction Module 262
Messages Module 260
Object Request Queuing Module 258

FIG. 2A

FIG. 1A

Thus, a POSITA would have found that the inventors lacked possession of these claimed ideas. Baker ¶¶ 69-79.

D. No Support for Receiving the Logical (Block) Address(es) of Claims 1 and 12

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 logical block address (instead of sending a whole file indication map). Ex. 1033; Ex. 1017-1018. Years later, the applicant claimed this feature.

Claim 1 of the ’727 Patent recites, “a message received from a host operating system, the message indicating that the identified logical address is erased."[3] Ex. 1001, 53:38-40. Claim 12 recites to "receive, from the storage client through the storage interface, an empty block directive command and a range of logical block addresses." *Id.*, 54:54-56. The 2006 Provisional lacks any disclosure of receiving these logical (block) address(es).

Although UTL looks to provisional "claim" 18 for support, 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 ¶ 82. Nowhere does the 2006 Provisional state that the "empty-block" hint/directive should indicate logical (block) address(es). *Id.* 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 includes logical (block) address(es), as opposed to identifying the blocks some other way. Thus, a POSITA would not have understood "claim" 18 of the 2006 Provisional to explicitly teach receiving logical (block) address(es). Baker ¶ 82.

[3] UTL disputes what is indicated as erased but seems to agree to the involvement of a logical address. *See infra* Section VI (construing claims).

Nor does the 2006 Provisional provide an inherent or implicit disclosure of the empty-block hint/directive comprising logical (block) address(es). Baker ¶¶ 83-84. In fact, issued claim 12 recites receiving the "empty-block directive" and separately, "logical block addresses." Ex. 1001, 54:55-56. Some commands use physical addresses, not logical addresses, to identify blocks. Ex. 1001, 14:63-15:3; 17:54-60, 21:20-26, 35:34-36. Thus, a POSITA would have recognized that a hint or directive might have identified a physical address instead of the claimed logical (block) address(es). Baker ¶ 84. Alternatively, a different instruction, separate from the hint or directive, could include the logical (block) address(es), like how erase commands worked in the SD Specification. *Id.* As another alternative, Jenett taught a solution by sending an entire "file indication map." Ex. 1033, *passim*.

Thus, provisional "claim" 18 fails to expressly, implicitly, or inherently disclose any of these options, much less the specific option of receiving logical (block) address(es) per claims 1 and 12 of the '727 Patent.

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 provisional "claim" 18, the only other provisional "claims" that mention an empty-block hint/directive include "claims" 14 and 29. Baker ¶ 87; Ex. 1015 at 35, 103. These provisional "claims" discuss entirely different problems and solutions, and neither discloses a hint/directive indicates or is received with logical (block) address(es). Baker ¶¶ 87, 94; Ex. 1015 at 35, 103. 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 ¶ 85. Provisional "claim" 14, titled "Object Based storage and Garbage Collection," also shows that the inventors had not solved how to integrate object storage and garbage collection. Ex. 1015 at 35 (reciting incomplete sentences as the "solution"); Baker ¶ 87.

Other passages in the 2006 Provisional mention a "logical block" in other contexts, but these parts also fail to describe receiving logical (block) address(es) as claimed. For example, "claims" 22 and 24 of the 2006 Provisional mention logical blocks *in a map* but say nothing about receiving logical (block) address(es) in or with the empty-block hint/directive. Baker ¶ 71; Ex. 1015 at 68, 72-74.

Additionally, to the extent that the UTL refers to provisional "claim" 30, this description uses an ObjectID as part of object-based commands—a type of command different from block commands such as the empty-block hint/directive. Baker ¶ 95; Ex. 1015 at 104. Also, this provisional "claim" directs the POSITA away from block-based commands by giving the POSITA "a killer reason to change from block access to object access." Ex. 1015 at 104; Baker ¶ 95. Provisional "claims" 31 and 32 similarly discuss the incompatible object-based system and provide no further detail of empty-block hint/directive. Ex. 1015 at 105-06; Baker ¶ 95.

Provisional "claims" 22 and 23 mention "messages," but these "messages" refer to "units on NAND flash," or in other words, "encapsulate raw data" stored on a media. Ex. 1015 at 68-69. "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 ¶¶ 89-90.

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 a file system or client. Baker ¶ 92.

The Patent Owner also relies on provisional "claims" 2, 9, and 28, but likely to support other claim elements. Ex. 1015 at 23, 31. Provisional "claims" 2, 9, and 28 say nothing about the empty-block hint/directive. *Id.*; Baker ¶¶ 93, 98. The Patent Owner also relies on provisional "claims" 13, 25, and 33 which relate to garbage collection, but likely only to support a different challenged element; thus, Section VI.F below addresses these. Ex. 1015 at 35, 74-75, 113. Provisional "claims" 13, 25, and 33 say nothing about the empty-block hint/directive. *Id.*; Baker ¶¶ 88, 91, 96.

Thus, the 2006 Provisional fails to describe receiving logical (block) address(es) 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 ¶¶ 81-98.

E. No Support for the Logical (Block) Address(es) Relating to Anything That "is erased," or "is deleted at the storage client" Per Claims 1 and 12.

The "empty-block" hint/directive of the 2006 Provisional discloses something different from '727 Patent claim 1's recitation of "the message indicating that the identified logical address is erased" and claim 12's recitation of indicating that "data corresponding to the received logical block address is deleted at the storage client." Ex. 1001, 53:38-40, 54:54-56, 54:61-64; *see also* Section VI, *infra* (explaining UTL's construction).

The 2006 Provisional describes the empty-block hint/directive as: "regarding which specific blocks do not hold data that needs to be preserved," "for the blocks that contained the data for that file, but no longer need to remember the contents," and possibly "incorporating a flag to indicate that data not only need not be preserved, but should be destroyed." Ex. 1015 at 40. The phrases "should be destroyed" and "need not be preserved" mean that the data is *not yet erased*. Baker ¶ 101. So, this provisional disclosure would have led a POSITA to understand that associated data *does not need to be preserved* or perhaps *should be destroyed*, but certainly not that anything "is erased," or "is deleted," as claimed. *Id*. Nor does the alternative solution, writing zeros to blocks "whose contents are no longer needed," support the claims that anything "is erased" or "is deleted." *Id*.

As discussed in Sections IV.D, IV.F, and IV.G, no other provisional "claim" relates to the same idea. Baker ¶ 102. Thus, a POSITA would have found that the inventors lacked possession of these claimed ideas. *Id.*

F. No Support for the Indexer or Logical-to-Physical Translation Layer's Responsive Actions in Claims 1 and 12.

Claim 1 includes an "indexer" that is configured to "remove an assignment between an identified logical address and a physical address." Claim 12 recites to "update the logical-to-physical translation layer to indicate that data ... corresponding to the received logical block addresses do not need to be preserved." The 2006 Provisional fails to further support an indexer or logical-to-physical translation layer performing these specifically claimed actions in response to receiving the logical (block) address(es). Baker ¶¶ 104-112.

For claim 1, the applicant arguably added support later, in Exhibit 1001, 51:50-52. The 2006 Provisional lacks a corresponding disclosure.

Provisional "claim" 18 states that an empty-block hint is "regarding which specific blocks do not hold data" and that a "file-system can issue an 'empty-block' directive for the blocks that contained data for that file." Ex. 1015 at 40. But this disclosure says nothing about the specifically claimed actions of removing assignments or updating the logical-to-physical translation layer. Baker ¶¶ 105-107.

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 ¶ 106. The lack of this teaching leaves the implementation to the imagination of a POSITA, including whether to preserve the data or whether to destroy it, as signified by the empty-block hint/directive. *Id.* ¶¶ 106-107. Regardless, the 2006 Provisional contains no disclosure of enhancing garbage collection by updating an indexer or logical-to-physical translation layer, as claimed.

Nothing in the disclosure of an empty-block hint/directive or garbage collection inherently or implicitly requires updating an indexer or logical-to-physical translation layer as claimed. Baker ¶ 107. Indeed, claims 1 and 12 recite alternative responses, neither of which is inherent: an indexer can "remove an assignment between an identified logical address," or update a logical-to-physical translation layer to "indicate that data ... do not need to be preserved." Ex. 1001. The 2006 Provisional fails to expressly, implicitly, or inherently disclose either option. *See also* Baker ¶ 107 (providing alternative actions).

UTL cannot salvage support from other parts of the 2006 Provisional. 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 ¶ 108. 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 provisional "claim" 25. Ex. 1015 at 40, 74; Baker ¶ 108. 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. Identifying old data in response to a different condition (when new data is appended) does not support removing an assignment from an index or updating of the logical-to-physical translation layer as recited in claims 1 and 12 of the '727 Patent. Baker ¶ 108.

Neither provisional "claims" 13 nor 33, which also relate to garbage collection, disclose performing any actions in response to the empty-block hint/directive. Ex. 1015 at 35, 113. Provisional "claim" 13 discusses a garbage collection snapshot system and, at best, vaguely references a "clear-block" directive on nonexistent page "xxx." *Id.* at 35. Provisional "claim" 33 teaches eliminating revision numbers used during garbage collection. *Id.* at 113. These provisional "claims" 13 and 33 show no possession of the claimed indexer or logical-to-physical translation layer's responsive actions. Baker ¶¶ 110-111.

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. Baker ¶ 109; 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 ¶ 109.

The other parts of the 2006 Provisional do not provide the missing support. 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 ¶ 109.

As discussed in Sections IV.D and IV.E, above, no other provisional "claim" relates to the same idea. Thus, a POSITA would have found that the inventors lacked possession of these claimed ideas. Baker ¶¶ 104-112.

G. No Support for Storing "Persistent Data" in Claim 12

The 2006 Provisional makes no mention of storing "persistent data on the flash memory device, the persistent data indicating that the data corresponding to the received logical block addresses is deleted at the storage client."

"Claim" 18 in the 2006 Provisional only discloses what the empty-block hint/directive signifies, not what to do afterward. Ex. 1015 at 40. Nothing discloses storing *persistent* data in response to the empty-block hint/directive, especially not persistent data indicating that other data "is deleted at the storage client." The lack of this detail is important because the 2006 Provisional teaches that a different data structure "may not be stored persistently at all." Ex. 1015 at 68.

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 ¶ 106. This omission leaves the implementation to a POSITA's imagination, including whether to preserve the data or whether to destroy it, as signified by the empty-block hint/directive. *Id.* ¶¶ 106-107. Regardless, the 2006 Provisional contains no disclosure of enhancing garbage collection by storing more, persistent data in response to the empty-block hint/directive. Baker ¶ 117.

Unconnected to the empty-block hint/directive, provisional "claim" 22 describes storing mapping data is stored in a block mapping table, but the block mapping table is in DRAM, which is volatile rather than persistent. Baker ¶ 114; Ex. 1015 at 68 (the mapping table "may not be stored persistently at all"). Moreover, none of this mapping data is disclosed as "indicating that the data corresponding to the received logical block addresses is deleted at the storage client," as claimed.

Thus, "claim" 18 of the 2006 Provisional fails to expressly, implicitly, or inherently disclose to "store persistent data" indicating anything about the received logical block addresses as recited in claim 12 of the '727 Patent.

UTL cannot salvage § 112 support from other parts of the 2006 Provisional for the same reasons discussed in Section IV.D-F, no other provisional "claim" relates to the same idea. Thus, a POSITA would have found that the inventors lacked possession of storing persistent data with the specific indication. Baker ¶¶ 114-117.

H. The Dependent Claims Lack Priority

Claims 2-5 and 14-16 lack priority support from the 2006 Provisional because of dependency. Claims 2 and 3 independently lack priority because the 2006 Provisional fails to describe any further details about any index entries, removing index entries, or maintaining index metadata in a storage controller. Baker ¶ 118. Claim 4 independently lacks priority because the 2006 Provisional fails to support that "the indexer comprises firmware of the solid-state storage controller." *Id.* Claim 5 independently lacks priority because, although the 2006 Provisional mentions improving garbage collection, the Dec. '06 Provisional fails to support doing so by designating anything in response to the empty-block hint/directive. *Id.* Claims 15 and 16 independently lack priority because the 2006 Provisional has no support for how to respond to read requests, especially requests to read data previously designated by the empty-block directive command. *Id.* ¶ 119. Moreover, provisional "claim" 18's "alternative" solution to "write all zeros" in flash memory does not support dependent claims 15 and 16 because (1) this "alternative" solution is performed *instead of* sending the empty-block command, and (2) *writing zeros to flash* does not support any claim about responding to a *read request*.

Thus, the 2006 Provisional disclosure would not have conveyed possession of these dependent claims to a POSITA, and priority to the 2006 Provisional "must be denied." Baker ¶¶ 118-119; 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 ¶ 59. 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. *Id.* ¶ 61. The POSITA would also have had a solid understanding of the various background concepts behind flash memory management. *Id.* ¶ 60 (listing concepts and references).

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.[4] The district court's constructions for the claims at issue in this Petition are:

Claim Term	Court	UTL
"[an] empty-block directive command" Claims 12, 15	A command that indicates that certain blocks contain data that does not need to be preserved.	A command that indicates that certain blocks contain data that do not need to be preserved.
"the identified logical address is erased" Claim 1	Plain and ordinary meaning.	The data identified by the logical address does not need to be preserved.
"logical to physical translation layer" Claims 12-14	Not indefinite; not subject to § 112(f); plain and ordinary meaning.	Not indefinite and not subject to § 112(f).

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

Claim Term	Court	UTL
"indexer" Claims 1-4	Not indefinite; not subject to § 112(f); plain and ordinary meaning.	Not indefinite and not subject to § 112(f).
"storage processor configured to … update … and store persistent data …" as claimed Claim 12	Not indefinite; not subject to § 112(f); plain and ordinary meaning.	Not indefinite and not subject to § 112(f).

Petitioners disagree with UTL's positions. As to "the identified logical address is erased," UTL asserts that its proposed construction is the plain and ordinary meaning. Because the invalidity arguments rely on the accused Trim command predating the priority date of the '727 Patent, the Board need not resolve any claim construction disputes and may assume UTL's constructions solely for this IPR.

IX. Precise Relief Requested

A. Proposed Ground 1

Claims 1-6 and 12-16 are rendered obvious by the Shu Patent (Ex. 1003) in view of the Shu Trim Proposal revisions 0-1 (Exs. 1017-1018), and Jenett (Ex. 1033) which incorporates Ban (Ex. 1025) by reference in its entirety. (Ex. 1033, 3:53-55, 5:24-26.) *See Husky Injection Molding Sys. Ltd. v. Athena Automation Ltd.*, 838 F.3d 1236, 1238, 1247-49 (Fed. Cir. 2016) (vacating board's refusal to allow incorporation by reference).

B. Proposed Ground 2

Claims 1-6 and 12-16 are rendered obvious by the Shu Patent in view of the Shu Trim Proposal revisions 0-1, Jenett, and Ban. Ground 2 differs from Ground 1 solely by combining Jenett and Ban under § 103 instead of Jenett incorporating the Ban reference.

C. Qualifying Prior Art

For the reasons discussed in Section VI, above, the challenged claims of the '727 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	Fig. 3 (showing computing device 301, processing

that includes at least one processor and memory;	unit 307, memory 309), [0028].
[1b] a file system; and	Fig. 1 (showing file system 112b), [0014], [0016]-[0017].
[1c] a solid state drive ("SSD") driver that, based on execution by the at least one processor, is configured to:	Fig. 1 (showing SSD driver 114b), [0012]-[0014], [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] ("such 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,* 800 F.3d at 1378. Only one claim from the later issued patent must be supported by the provisional. *See Cysco 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 were published in April 2007 (rev. 0) and July 2007 (rev. 1) thereby qualifying as prior art under § 102 (a). Ex. 1039; Ex. 1010 § 8.1.2;

Ex. 1017; Ex. 1018; Ex. 1039. T13 published the proposals on its freely accessible website T13.org, and industry representatives met to discuss the Trim Proposals. 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 described in the Shu Patent.

Ban issued in 1995 and qualifies as prior art under §§ 102(a), (b), and (e). Ex. 1025. Jenett issued in 2000 and qualifies as prior art under §§ 102(a), (b), and (e). Ex. 1033.

D. Jenett Incorporates Ban by Reference

A host document incorporates material by reference if it "identif[ies] with detailed particularity what specific material it incorporates and clearly indicate[s] where that material is found in the various documents." *Husky Injection*, 838 F.3d at 1248. Courts assess whether a skilled artisan would understand the host document to describe with sufficient particularity the material to be incorporated. *Id.* A blanket statement incorporating an entire document by reference is sufficient. *Harari v. Lee*, 656 F.3d 1331, 1335 (Fed. Cir. 2011) (holding a reference is "in its entirety," even if other incorporation language refers to specific portions).

Jenett incorporates Ban (U.S. Pat. No. 5,404,485) in its entirety. Ex. 1033, 3:53-55, 5:24-26. Thus, the Board should treat Ban as incorporated by Jenett.

Aside from the blanket incorporation, a POSITA would have understood that

Jenett intended to incorporate at least all parts of Ban relating to the flash translation layer (FTL), operations between memory cards and an operating system, the file management system including file deletion, and the operation of flash memory. Ex. 1003, 1:46-56, 3:45-60, 4:65-5:36; Baker ¶ 179. This includes all parts of Ban relied on herein.

E. The Proposed Grounds Are 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 the Patent Owner 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 2007 Provisional, the Examiner had no reason to look at the sixth revision of the Trim Proposal before allowing the application. So although the Examiner considered U.S. Publication 2008/0263305 to Shu et al., the Examiner did not do so in view of the earlier Shu Trim Proposals, and Jenett/Ban as proposed herein.

The grounds for trial presented in this Petition are not cumulative to issues already examined in a parallel IPR proceeding, IPR2021-00345. 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 (Nov. 2019).

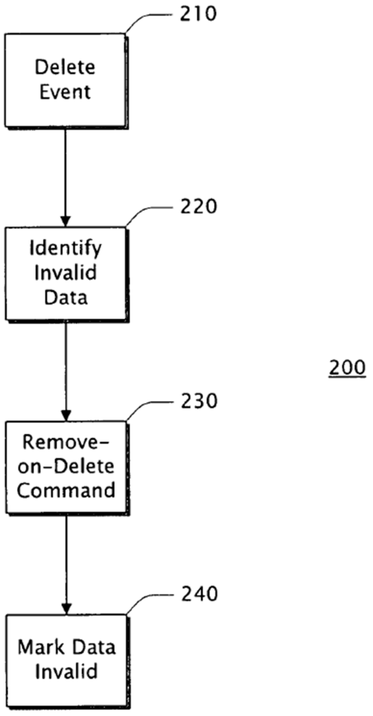
The *General Plastic* factors do not warrant denying this Petition. *Gen. Plastic Indus. Co. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (P.T.A.B. Sept. 6, 2017). IPR2021-00345 relates to overlapping claims of the '727 Patent and was filed on December 22, 2020. At that time, investigation into the Shu Trim Proposals remained ongoing, and Petitioners did not uncover relevant materials, such as Frank Shu's presentations (Exs. 1022, 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 UTL from unfairly using hindsight to capture Frank Shu's invention.

X. The Prior Art

A. Summary of the Shu Patent

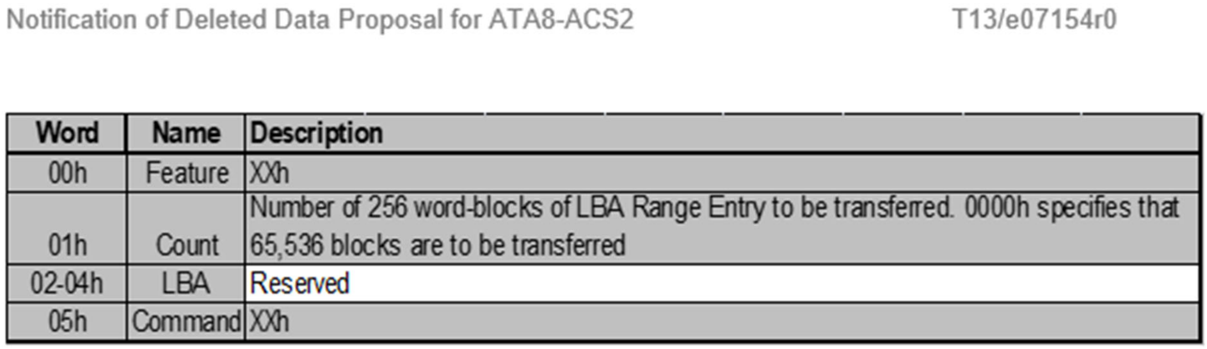
The Shu Patent relates to managing SSDs with flash memory. Ex. 1003, 1:12-

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 ¶¶ 170-171; Exs. 1017-1018. UTL now accuses Trim of infringement. Exs. 1017-1018. The 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.



Notification of Deleted Data Proposal for ATA8-ACS2 T13/e07154r0

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 into the Data Set Management (DSM) command. Ex. 1018 § 6.1.1.

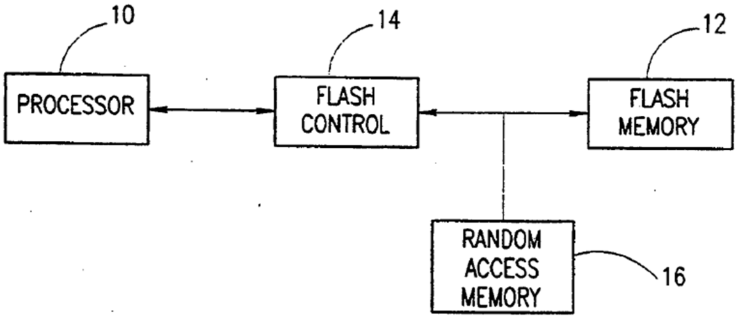
C. Summary of Ban

The industry credits Ban's 1995 patent as the invention of the modern flash translation layer, or "FTL." Ex. 1020 § 2.2 (citing "Ban [5]"); Ex. 1033, 1:50-53, 5:21-26 ("A well-known flash translation layer is disclosed … in U.S. Pat. No. 5,404,485 issued in 1995."); Baker ¶ 242. The 2006 Provisional admits that patented FTL's assigned to M-Systems, such as Ban, had become one of the "[t]raditional flash storage systems." Ex. 1015 at 77; Ex. 1025, cover (showing assignment).

In Ban's flash memory system, "[a] table, called a virtual map, converts virtual addresses to physical addresses." Ex. 1025, 2:6-8. The "virtual" address refers to the "computer generated address." *Id.*, 2:31-32. POSITAs also call this a "logical" address. Baker ¶ 181. Ban's map also tracks which data are "deleted and not writable." Ex. 1025, 4:47-50, 5:45-46, 5:64-66, 8:7-9.

This petition relies on Ban for showing the general components of a generic flash memory device that manages logical-to-physical mappings. Ban Figure 1 shows that components include a flash memory controller and flash memory:

FIG.1



D. Summary of Jenett

The year after Ban issued, Jenett realized that Ban's system could be improved if the FTL avoided preserving deleted files. Ex. 1033, cover, abstract, background. Jenett identifies Ban's system with an FTL as the relevant background art to improve on. *Id.*, 1:50-53. Jenett teaches that an operating system sends a file indication map to the FTL, and the FTL compares the file indication map to an earlier version of the file indication map to identify which blocks are invalid, permitting erasure of those blocks. *Id.*, abstract, Fig. 4.

Jenett's figure 4 shows the "FTL structures," including the BAM (a block allocation map) stored in the flash memory card. Jenett's BAM "is a physical to virtual map which associates particular physical sectors of the flash medium with a related virtual address, provided that a relationship exists," just like Ban's FTL. Ex. 1033, 6:16-18. Jenett also teaches marking deleted data as "invalid." *Id.*, 3:11-13, 5:16-21, Fig. 3 block 404.

Jenett teaches various techniques for marking data invalid. One marking technique includes making "the blocks associated with each deleted file invalid in the block allocation map." *Id.*, 4:56-57. As another marking technique, "block allocation map 501a(1) is updated to delete the association between the physical location at which the identified deleted file was stored and the virtual address formerly connected with the particular physical location." *Id.*, 6:12-16.

E. Motivation to Combine

The Shu Patent provides an explicit teaching to modify existing flash devices, such as Jenett's or Ban's flash memory systems, to use Shu's newly proposed remove-on-delete (Trim) command. The Shu Patent teaches to implement "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 Shu Patent teaches that 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.

Jenett and Ban provide examples of solid-state, flash memory devices that would benefit from receiving the new remove-on-delete command taught in the Shu Patent. Thus, a POSITA would have used a flash memory device, like the flash memory device from Jenett or Ban, to receive the remove-on-delete command. Baker ¶¶ 183-185. Jenett already had the same idea of identifying invalid data, albeit by sending a file indication map instead of a remove-on-delete/Trim command. *Id.* ¶ 184; Ex. 1033. Identifying invalid data improves the performance by avoiding "unnecessarily operating on invalid data," such as during merge operations. Ex. 1003, 3:8-17, 4:29-34. In Ban, data is merged or transferred from an old block into a new block before erasing an old block. Ex. 1025, 2:61-3:2; Baker ¶ 208. Both the Shu Patent and Jenett address this problem. Ex. 1003, 3:8-15; Ex. 1033, 1:12-21. Shu's remove-on-delete/Trim command identified the invalid data more efficiently than Jenett by specifying the starting logical block address and length of invalid data rather than sending an entire file indication map. Ex. 1003, 3:17-22; Ex. 1017 § 3; Baker ¶ 185.

Indeed, 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 SSDs. Frank Shu proposed his new Trim command to representatives of major companies in the flash storage industry. Ex. 1010 §§ 3.3, 8.1.2. Microsoft announced that the Windows operating system would support the command for SSDs. Ex. 1022 at 8.

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 T13, the organization 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 flash memory devices like those described in Jenett and Ban. Baker ¶¶ 183-185. 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.* ¶¶ 186-187.

XI. Ground 1: Obvious Over the Shu Patent, the Shu Trim Proposals, and Jenett, Which Incorporates Ban.

UTL accuses the Trim command 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 provided written description support for the asserted claims. Thus, Frank Shu's Trim Proposals, combined with details of a generic flash memory device (e.g., as shown in Jenett/Ban), invalidate the claims under the UTL's interpretation of the claims.

Petitioners allege in co-pending litigation that accused products that implement the TRIM command do not infringe, but for purposes of this IPR petition, 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].[5]

The Shu Patent discloses an apparatus in the form of a computer system with a solid-state storage device ("SSD"). Ex. 1003, Figs. 1 (cropped, reproduced below left), 3 (reproduced below); Ex. 1002,[6] Figs. 1, 3; Baker ¶¶ 189-190.

[5] The appended Claim Listing provides the claim language.

[6] Citations to Exhibit 1002 show the Shu Patent's priority to its provisional.

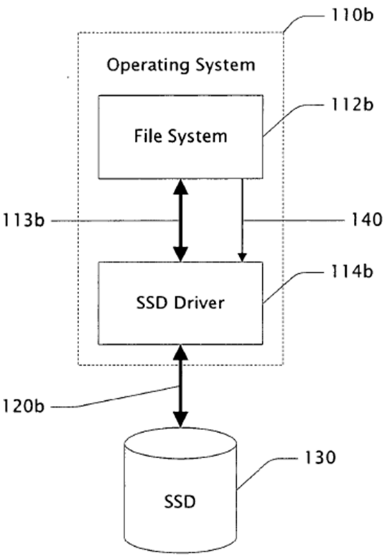


FIG. 1

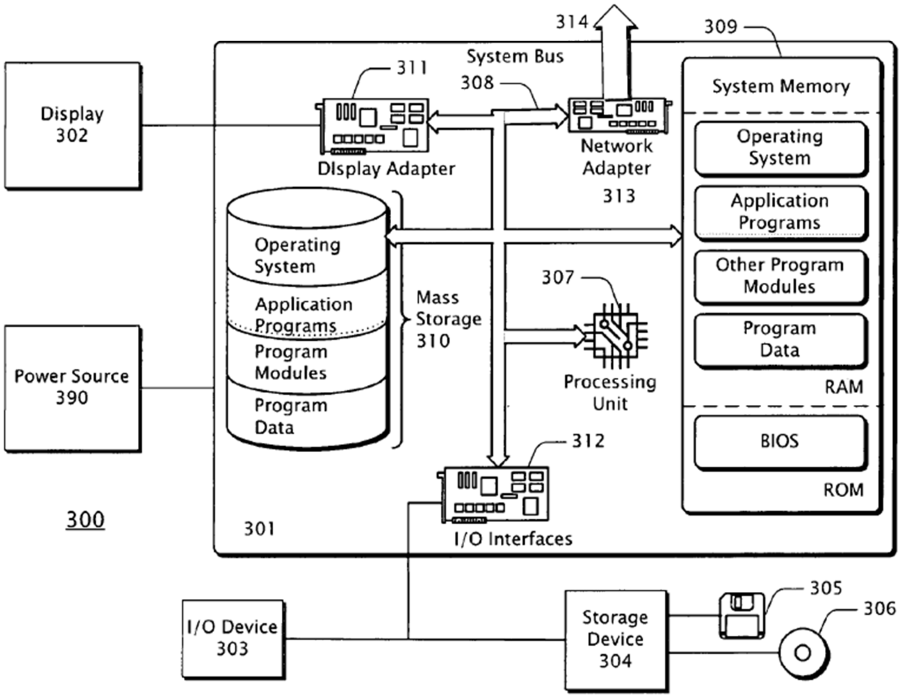


FIG. 3

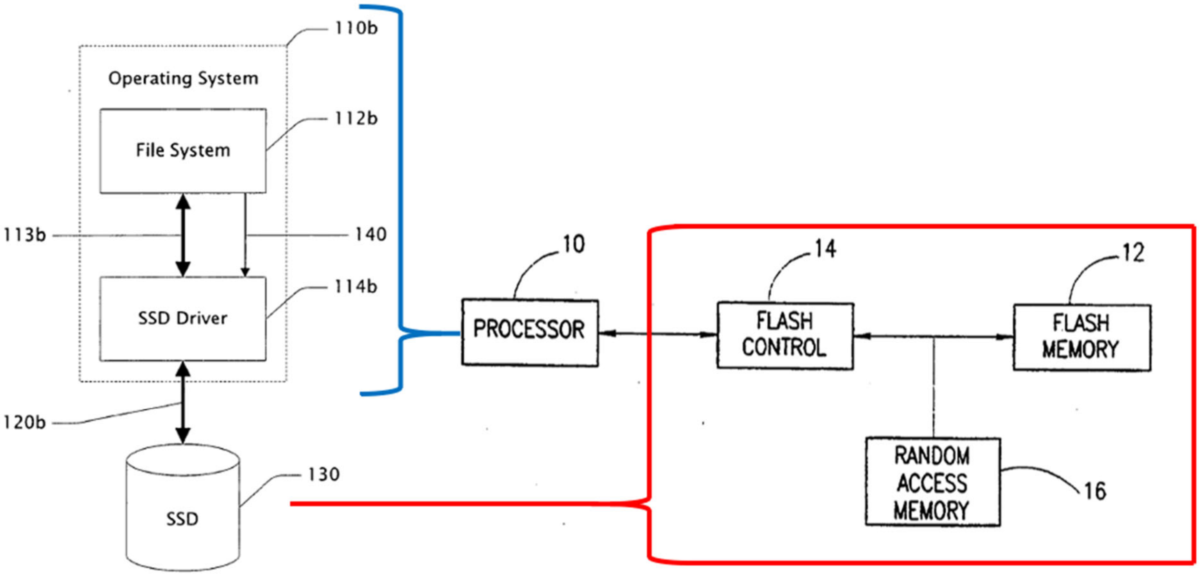
b) Element 1[b].

The Shu Patent shows a solid-state storage medium in the form of the SSD in figure 1 (reproduced above). Ex. 1003, 1:13-16, 2:52-57, Fig. 1, Fig. 3 (showing "storage device"); Ex. 1002, [0001], [0013], Fig. 1, Fig. 3; Baker ¶¶ 192-193.

c) Element 1[c].

The Shu Patent shows that the SSD receives requests from the computer "Operating System" of figure 1 "to store and access data on SSD 130." Ex. 1003, 2:44-49, Fig. 3 (showing computer system 301 and storage device 304); Ex. 1002, [0012], Fig. 3. Thus, a POSITA would have understood that storage operations on the SSD occur in response to requests from a computer system. Baker ¶ 194.

The Shu Patent does not detail the components inside the SSD, but a POSITA would have known that SSDs at the time included a flash memory controller, flash memory, and random access memory as shown in Figure 1 of Ban. Ex. 1025, Fig. 1; Ex. 1001, Fig. 1; Baker ¶ 195 (illustrating annotated combination, reproduced below); *see also supra* Section IV (explaining the basic components); Ex. 1033, 5:21-26.



From Shu Patent, Fig. 1

From Ban, Fig. 1

Ban shows the generic components in Shu's SSD

Ban's storage processor runs an operating system that includes a file system that sends requests to "write[] data to, and read data from, a flash memory," which the flash memory controller receives and implements using a memory map that "converts virtual[7] addresses to physical addresses." Ex. 1025, 1:52-67, 2:1-8, 3:56-4:5. Thus, a POSITA would have found it obvious that Shu's SSD included a flash memory controller, which is "a solid-state storage controller configured to implement storage operations on the solid state storage medium in response to requests from a computer system," as claimed. Baker ¶ 195.

[7] Ban's "virtual" addresses are "logical" addresses. Ex. 1025 at 4:51-59 (computer-generated logical unit numbers are "interpreted by the flash controller 14 as virtual addresses."); Baker 181.

d) Element 1[d].

The Shu Patent does not detail the inner components of the SSD, but a POSITA would have known that a storage processor "writes data to . . . a flash memory 12 in blocks at specific address locations," as explained by Ban and incorporated in Jenett. Ex. 1025, 3:60-4:10, Fig. 1; Ex. 1033, 5:21-26; Baker ¶ 196. A flash memory controller receives and processes these data storage operations. Ex. 1025, Fig. 1; Baker ¶ 196. During these write storage operations, the controller maps a computer-generated, virtual address to a physical memory block address. Ex. 1025, 2:1-8, 2:20-34, 3:31-39; Ex. 1033, 6:16-19. A POSITA would have interpreted the computer-generated, "virtual address" as the claimed "logical address of a logical address space." Baker ¶ 198. Because the map locates and assigns new, unwritten physical block addresses for each virtual address, data pertaining to each virtual address is stored "at respective physical addresses of the solid-state storage medium," as recited in the claim. *Id.*; Ex. 1025, 2:37-40.

For the reasons explained in Section IX.A.c (Element 1[c]), above, Ban as incorporated in Jenett describes the generic operations within the Shu Patent's SSD. The Shu Patent and Shu Trim Proposals specifically envisioned that the SSD would use mappings like those disclosed in Ban. Ex. 1003, 4:51-58; Ex. 1002, [0022]; Ex. 1017 § 2 ("A[] device will further remap LBA to its internal page and block for SSD."). Thus, it would have been obvious to a POSITA that the Shu Patent's SSD would use a storage controller for "storing data pertaining to logical addresses of a logical address space at respective physical addresses of the solid-state storage medium," as claimed. Baker ¶ 199.

e) Element 1[e].

UTL accuses the FTL of infringing this element. Ex. 1013 at 3. The industry credits Ban's patent as teaching the modern FTL, and the 2006 Provisional admits this. Baker ¶¶ 142, 200; *see supra* Section X.C. POSITAs also refer to the FTL as an "index," "map," or "mapping." Baker ¶ 200; Ex. 1013 at 3 (accusing "a map"); *see also* Ex. 1025, 1:66-67, 2:1-8, 2:31-33, 3:64-4:5, 4:51-61, 4:67-5:17, Fig. 4.

As explained in Section XI.A.d, above, the Shu Patent and Trim Proposals envisioned sending the remove-on-delete/Trim command to an SSD that uses mappings, referring to a system like Jenett that incorporates Ban's FTL for "storing data pertaining to logical addresses of a logical address space at respective physical addresses of the solid-state storage medium." Ex. 1003, 4:51-58; Ex. 1002, [0022]; Ex. 1017 § 2. Jenett's system uses a block allocation map, which is "a physical to virtual map which associates particular physical sectors of the flash medium with a related virtual address." Ex. 1033, 6:16-19; *see also* Ex. 1025, 2:31-38 (explaining that the map converts a computer-generated virtual address into the physical address being used to indicate a next unwritten block). Thus, a POSITA would have found it obvious that Ban's FTL/mapping was "configured to assign logical addresses of the logical address space to physical addresses in use to store data pertaining to the logical addresses on the solid-state storage medium." Baker ¶ 202.

UTL accuses the indexer of being "circuitry, software, and/or firmware" configured to assign logical addresses to physical addresses. Ex. 1013 at 3. Ban's flash controller is a circuit that manages the "correspondence between the virtual address space and physical address space," and "controller 14 functions may be carried out in software, firmware or hardware." Ex. 1025, 4:7-9, 4:56-61. Thus, a POSITA would have further found the obvious under UTL's interpretation. Baker ¶ 203.

Because the Shu Patent and Shu Trim Proposals envisioned using an SSD employing mapping as taught by Ban as incorporated in Jenett, this claim element, as interpreted by UTL, would have been obvious to a POSITA. Baker ¶ 204.

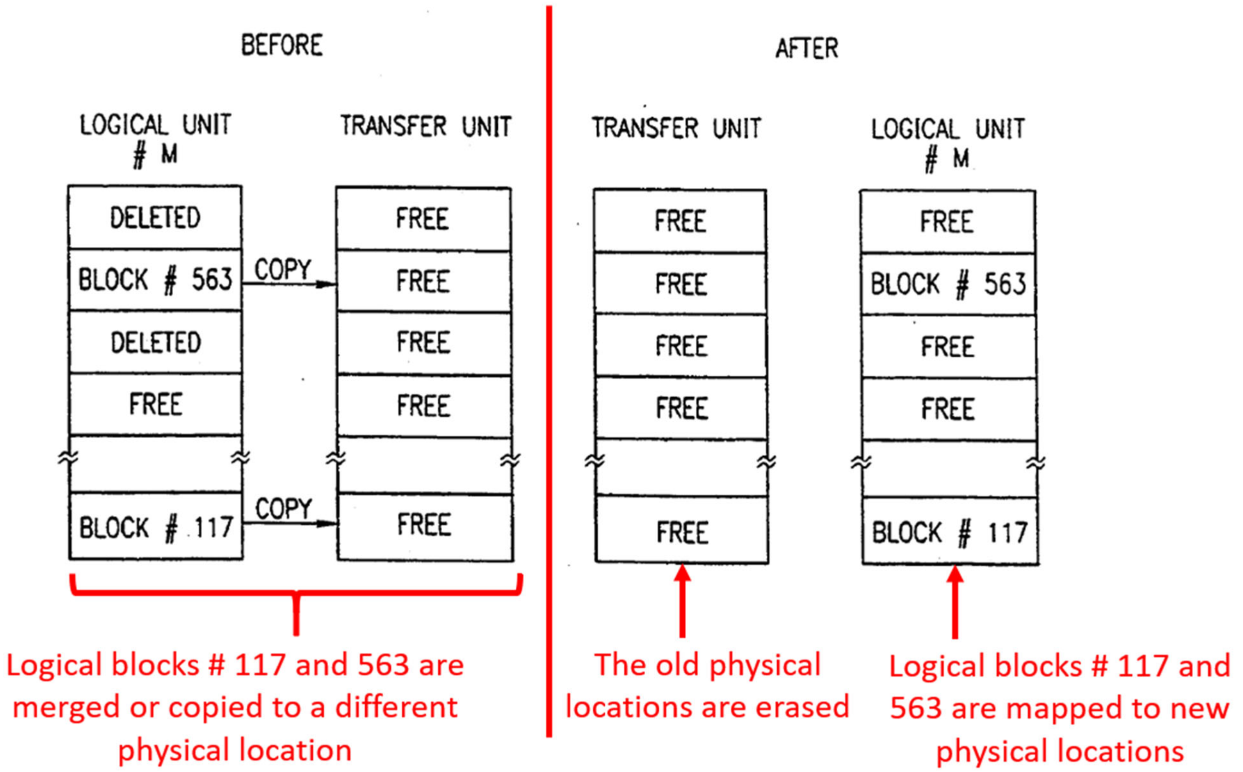
f) Element 1[f].

UTL accuses the Trim command of infringing the "message" element. Ex. 1013 at 3. Because the '727 Patent lacks priority to 2006, the accused command is prior art. UTL appears to accuse the FTL of infringing "remove an assignment between an identified logical address and a physical address," because "the FTL can remap the logical address to a different physical address," after a trimmed block is garbage collected and erased. *Id.* Ban taught the FTL with the accused remapping in 1995. *E.g.*, Ex. 1025, 4:59-66; 6:23-30 ("the logical to physical address map is changed" when a selected unit is erased), 8:30-49 (claiming the accused remapping).

The Shu Patent teaches to send the accused remove-on-delete/Trim command to an SSD. Ex. 1003, abstract, 1:60-63, 3:20-22, 4:35-38, 4:65-5:4, Fig. 2 block 230; Ex. 1002 at 23, [0013], [0014], [0020], [0023], Fig. 1, Fig. 2. This message is "received from a host operating system," as claimed. Ex. 1003, 3:62-4:3, Fig. 1; Ex. 1002, [0016], Fig. 1.

The remove-on-delete/Trim command "instructs the SSD . . . to mark the indicated data as invalid." Ex. 1003, 4:65-5:1; Ex. 1002, [0023]. "Once the deleted data is marked as invalid, the SSD device is freed from performing any operations to preserve, maintain … invalid data." Ex. 1003, 5:11-13; Ex. 1002, [0024]. Specifically, "merge" operations on invalid data can be avoided. Ex. 1003, 1:20-25, 3:35-40, 4:29-31; Ex. 1002, [0001], [0014], [0019]. A POSITA would have understood that "merge" operations refer to merging/copying data from old blocks into new blocks, before erasing the old blocks, as illustrated in Ban Figures 7 and 8. Baker ¶ 208; Ex. 1025, 6:9-27, Fig. 7 (annotated and reproduced below), Fig. 8.

FIG.7

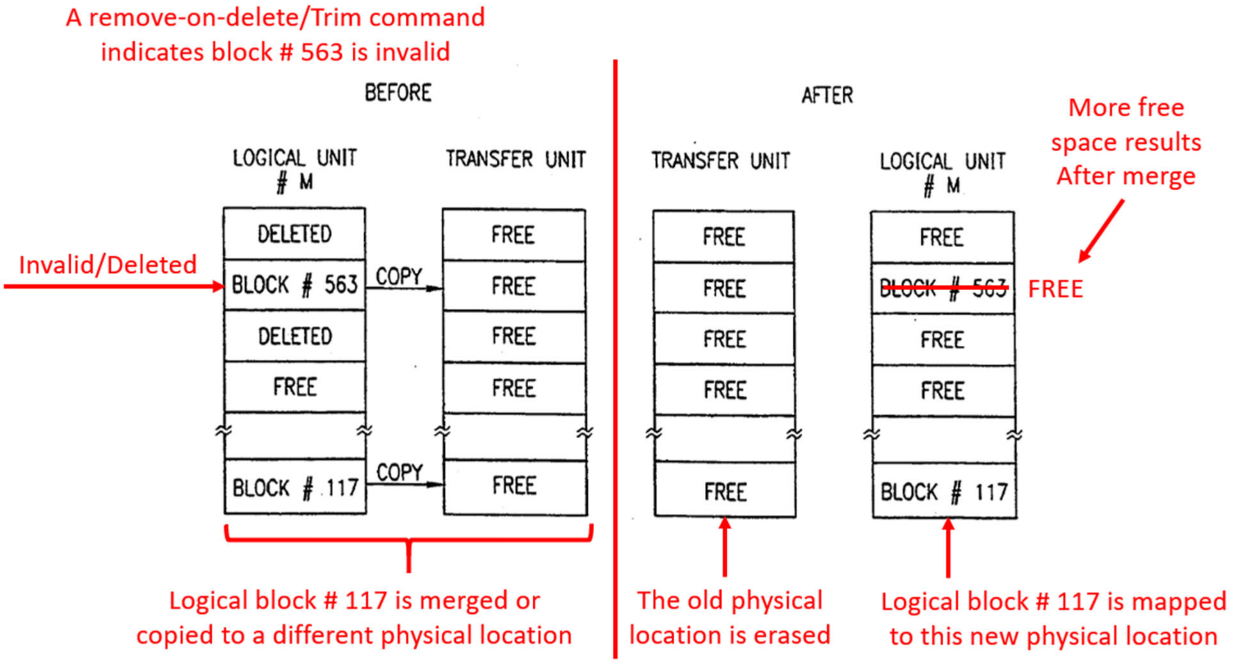


Logical blocks # 117 and 563 are merged or copied to a different physical location

The old physical locations are erased

Logical blocks # 117 and 563 are mapped to new physical locations

An SSD may not be aware that a particular location (e.g., block #563 pictured above) is invalid. Ex. 1003, 1:38-43; Ex. 1002, [0001]. Thus, the Shu Patent and Shu Trim Proposals propose using the remove-on-delete/Trim command to inform the SSD that a block (e.g., block #563), contains invalid data that does not need to be copied over. Baker ¶¶ 207-210; Ex. 1003, 4:26-31; Ex. 1002, [0019]. The resulting process is shown below:


A remove-on-delete/Trim command indicates block # 563 is invalid
BEFORE
AFTER
LOGICAL UNIT # M
TRANSFER UNIT
TRANSFER UNIT
LOGICAL UNIT # M
More free space results After merge
DELETED
BLOCK # 563
DELETED
FREE
BLOCK # 117
COPY
COPY
FREE
FREE
FREE
FREE
FREE
Invalid/Deleted
FREE
FREE
FREE
FREE
FREE
FREE
~~BLOCK # 563~~
FREE
FREE
FREE
BLOCK # 117
Logical block # 117 is merged or copied to a different physical location
The old physical location is erased
Logical block # 117 is mapped to this new physical location

Baker ¶ 209. Then, the mapping is updated accordingly. Ex. 1025, 6:23-29, 8:47-49, Fig. 8 (step 64). Thus, by applying UTL's interpretations, a POSITA would have found this element obvious because both the accused Trim command and Ban's FTL predate the '727 Patent. Baker ¶¶ 205-210.

The claim is also obvious under an alternative interpretation of "remove an assignment." Jenett taught to identify invalid data in order "to ensure that files deleted during normal computer operation are not unnecessarily preserved during file cleanup of a flash memory card array and medium," again referring to the same merge process referenced in the Shu Patent. Ex. 1033, 1:65-67; Baker ¶ 212. Jenett discloses a common technique for indicating invalid data: to "delete the association between the physical location at which the identified deleted file was stored and the virtual address formerly connected with the particular physical location." Ex. 1033, 6:13-16; *see also id.*, 6:20-33 (providing additional detail), 4:56-57, Fig. 3 block 404.

The Shu Patent's remove-on-delete/Trim command "instructs an SSD device … to mark the indicated data as invalid," and "[s]uch a mark may take any form sufficient to identify the invalid data to the SSD device." Ex. 1003, 4:65-5:1, 5:5-7; Ex. 1002, [0023], [0024]. Jennett provided a known way to do this, so a POSITA would have found it obvious for an SSD to delete an association between a logical address and a physical address to invalidate that association in response to a remove-on-delete/Trim command. Baker ¶ 213. Thus, a POSITA would have found it obvious for the indexer in the Shu Patent's SSD to remove an assignment between a logical block address (LBA) specified in the remove-on-delete/Trim command and a physical address of the solid-state storage medium in response to the remove-on-delete/Trim command received from a host operating system, when applying UTL's interpretation of "indexer" and "message." Baker ¶¶ 212-213.

g) Element 1[g].

UTL accuses the Trim command of being a "form of message which indicates that the identified logical address is erased." Ex. 1013 at 3. Because the '727 Patent lacks priority to 2006, the accused command is prior art, regardless of how UTL construes this term.

The remove-on-delete/Trim command specifies the logical block address.

Ex. 1003, 3:17-20; Ex. 1002, [0014]; Ex. 1017 § 3; Ex. 1018 § 6.1.1. The logical block address is identified to the SSD when "[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, 3:11-15. 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, such as by removing a reference to the deleted file from a directory or the like. Further, the file system may mark the data representing the file on the SSD as invalid. 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. The LBA or data location information describing the data to be deleted is typically sent by command via an interface to the SSD device. Upon receiving the command and associated data location information, the SSD and/or its driver can mark as invalid data stored on the SSD that corresponds to the deleted file.

Id. 1003, 3:8-25, 3:48-53; *see also id.*, 1:32-34, 1:55-63, 4:35-38, 5:3-11; Ex. 1002, [0014]-[0015]; *see also id.*, [0001], [0003], [0020], [0023]-[0024]. Although this quote does not explicitly refer to the remove-on-delete/Trim command as including the LBA, a POSITA would have seen this clarified in the Shu Trim Proposals. Ex. 1017 § 3; Ex. 1018 § 6.1.1; Baker ¶ 215. The SSD then processes the LBA as described above in Section XI.A.f. Thus, when applying the UTL's interpretation, this claim element would have been obvious to a POSITA. Baker ¶ 216.

B. Claim 2

a) Element 2[a].

For the reasons discussed in Section IX.A.e-f (elements 1[e]-[f]), above, a POSITA would have found it obvious that the Shu Patent's SSD includes an indexer that assigns logical addresses to physical addresses. The POSITA would have further found it obvious that an indexer does so using index entries because the virtual to physical memory map described in Jenett and Ban is a "table," where each row that maps a virtual to physical address is considered an index entry. Baker ¶ 218; Ex. 1025, 2:20-24.

b) Element 2[b].

The remove-on-delete/Trim command specifies a logical block address. Ex. 1003, 3:17-20; Ex. 1002, [0014]; Ex. 1017 § 3; Ex. 1018 § 6.1.1. This command "instructs the SSD device … to mark the indicated data as invalid," and "[s]uch a mark may take any form sufficient to identify the invalid data to the SSD device." Ex. 1003, 4:65-5:1, 5:5-7; Ex. 1002, [0023], [0024]. Jenett discloses a common technique used by SSDs to indicate invalid data: to "delete the association between the physical location at which the identified deleted file was stored and the virtual address formerly connected with the particular physical location." Ex. 1033, 6:13-16; *see also id.* 6:20-33 (providing additional detail), 4:56-57, Fig. 3 block 404. As discussed in Sections 1[f]-[g], above, a POSITA would have found it obvious to mark the data at the logical block address invalid by using Jenett's technique of deleting the association between the physical location and the virtual address, which refers to deleting an index entry, in response to the remove-on-delete/Trim command that identifies the LBA. Baker ¶ 219.

C. Claim 3.

According to UTL, "Logical block addresses and physical addresses are both forms of index metadata." Ex. 1013 at 4.

As previously discussed, all the asserted references teach an index or mapping of logical addresses and physical addresses. Ex. 1003, 1:34-39, 3:17-22; Ex. 1002, [0001], [0014]; Ex. 1033, 6:11-20; Ex. 1025, 2:6-8, 2:22-43, 4:51-66, cols. 5-12, Fig. 4. Ban teaches that the memory controller manages the address space and interprets the address correspondences. Ex. 1033, 3:65-67, 4:54-59. Thus, a POSITA would have understood that Ban's system uses index metadata maintained in the memory of this memory controller, when applying UTL's interpretation of "index metadata." Baker ¶ 222.

For the reasons discussed in Sections IX.A.c and e (elements 1[c], 1[e]), above, a POSITA would have understood that the SSD in the Shu Patent contained the system described in Ban as incorporated by Jenett, which includes a memory controller configured to assign logical addresses and physical addresses. Further, when applying UTL's interpretation, a POSITA would have found it obvious that these assignments use index metadata maintained in a memory of the memory controller of the SSD. *Id.* ¶ 222.

D. Claim 4.

For the reasons discussed in Section IX.A.e (element 1[e]), above, the indexer comprises firmware of the solid-state storage controller. Ban teaches that the "controller 14 functions may be carried out in software, firmware or hardware." Ex. 1025, 4:7-9, 4:56-61. Thus, a POSITA would have found this claim obvious. Baker ¶¶ 226-227.

E. Claim 5.

The 2006 Provisional admits that "garbage collection" was "commonly used" with flash memory systems. Ex. 1015 at 40. The Shu Patent envisioned that garbage collection would occur because the Shu Patent repeatedly references the "merge" step of garbage collection, where valid data is merged from an old block into a new block before erasing the old block. Baker ¶ 229; Ex. 1003, 1:20-25, 3:35-40, 4:29-31, 10:22-26; Ex. 1002, [0001], [0014], [0019]; *see supra* Section 1[f] (illustrating merging); Ex. 1025 at 2:61-3:2 (describing the merging and erasing part of garbage collection).

Specifically, the Shu Patent taught that, in response to receiving the remove-on-delete/Trim command, the flash device should "mark the indicated data as invalid." Ex. 1003, 4:65-5:1. This "marking may involve marking ... blocks or the like invalid." Ex. 1003, 5:7-8; Ex. 1002, [0023]. Thus, a POSITA would have understood that the "physical addresses previously assigned to the identified logical addresses" would be marked or "designated," as claimed. Baker ¶ 230. Moreover, "[o]nce the deleted data is marked as invalid, the SSD device is freed from performing any operation to preserve, maintain, or the like the invalid data." Ex. 1003, 5:11-13; Ex. 1002, [0024]. Thus, a POSITA would have found it obvious to designate that "the data is suitable for removal from the solid-state storage medium in response to the message," as claimed. Baker ¶ 230. A POSITA would have found it obvious that a "garbage collector" would have performed the marking because garbage collection was admittedly commonly used for this purpose. *Id.* Thus, when applying the UTL's interpretation, this element would have been obvious to a POSITA. *Id.*

F. Claim 6.

This element is taught by the Shu Patent and is inherently obvious from the Shu Trim Proposals, which are proposals for the ATA standard to support the Trim command. The Shu Patent shows an interface to the SSD in figure 1, and the Shu Patent explains that examples of the interface "include the advanced technology attachment ('ATA') interface." Ex. 1003, 2:49-50, 4:18-20; Ex. 1002, [0012], [0018]. As shown and discussed in Section IX.A.c (element 1[c]), above, this ATA interface couples to the storage controller. The Shu Trim Proposals are titled as "Proposal for ATA8-ACS2," meaning that they propose the Trim command for inclusion in the ATA8 reversion of the ATA standard. Exs. 1017-1018. Thus, this claim was obvious to a POSITA. Baker ¶¶ 232-233.

G. Claim 12

a) Element 12[a].[8]

The Shu Patent discloses the claimed system that includes a solid-state drive. Ex. 1003, Fig. 1; Ex. 1002, Fig. 1; *see also* Ex. 1017, *passim*; Baker ¶¶ 235-236.

b) Element 12[b].

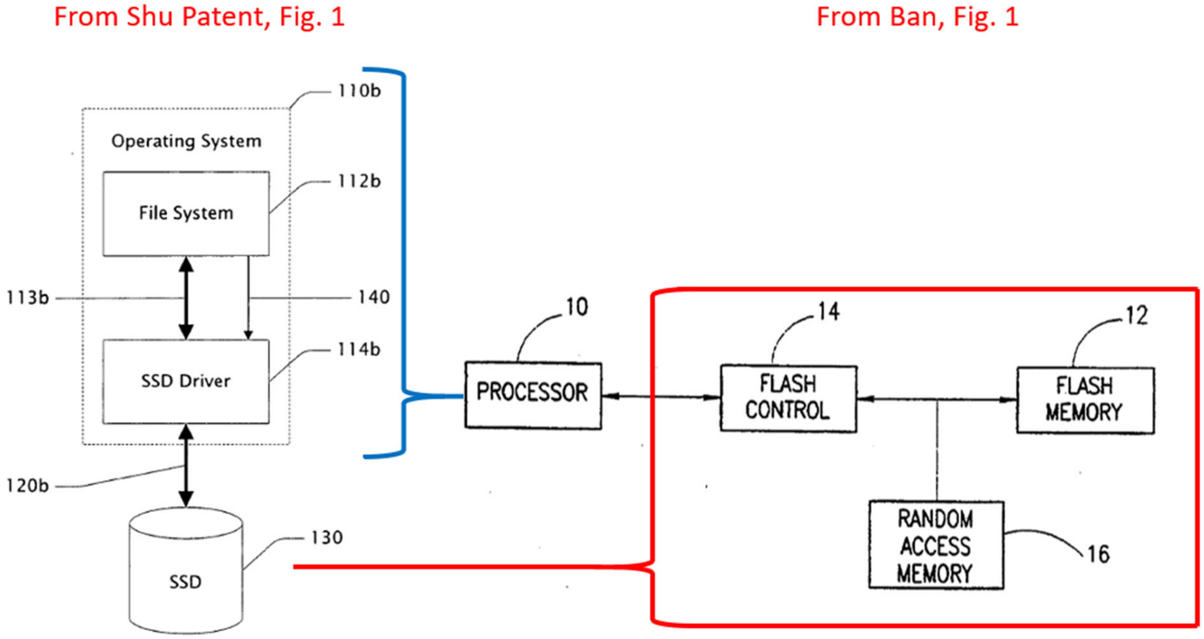
The Shu Patent shows that the SSD receives communications from the client "Operating System" via interface 120. Ex. 1003, 2:35-51, Fig. 1 (showing interface 120), Fig. 3; Ex. 1002, [0012], Fig. 1, Fig. 3. These communications include the remove-on-delete command, which is also referred to as the Trim command. Ex. 1002; Ex. 1003; Exs. 1017-1018; Baker ¶¶ 237-238.

c) Element 12[c].

The Shu Patent does not detail the inner components of the SSD, but a POSITA would have known that SSDs at the time generally included a flash controller, flash memory, and random access memory ("RAM") as shown in Figure 1 of Ban. Ex. 1025, Fig. 1; Ex. 1001, Fig. 1; *see also supra* Section VI

[8] The appended Claim Listing provides the claim language.

(explaining the basic components); Baker ¶ 239 (illustrating annotated combination, reproduced below).



Ban shows the generic components in Shu's SSD

A POSITA would have understood that Ban's flash controller is the claimed "storage processor" that interfaces with the processor 10 that runs the operating system. Baker ¶ 240; Ex. 1025, 3:52-54, 4:1-2, Fig. 1. The claimed "flash memory device" is flash memory 12, which is coupled to the flash controller. Ex. 1025, Fig. 1. Thus, a POSITA would have found these elements obvious because they were the generic components in the SSD. Baker ¶ 240.

d) Element 12[d].

UTL accuses the FTL of infringing this element. Ex. 1013 at 6. The industry credits Ban's patent as teaching the modern FTL, and the 2006 Provisional admits this. *Id.* ¶ 242; *see supra* Section X.C. POSITAs also refer to the FTL as an "index," a "map," or a "mapping." Baker ¶ 242; Ex. 1013 at 6 (accusing "a map"); *see also* Ex. 1025, 1:66-67, 2:1-8, 2:31-33, 3:64-4:5, 4:51-61, 4:67-5:17, Fig. 4 (showing both forward and reverse logical-to-physical mapping tables).

The Shu Patent and Shu Trim Proposals specifically envisioned that the SSD would use mappings like those disclosed in Ban's FTL, as incorporated by Jenett. Ex. 1003, 4:51-58; Ex. 1002, [0022]; Ex. 1017 § 2 ("A[] device will further remap LBA to its internal page and block for SSD."). Jenett uses the embodiment of Ban's FTL that maps logical block addresses to corresponding respective physical block addresses. Ex. 1033, 6:15-20; Ex. 1025, 1:65-67, 2:1-8, 3:64-4:5, 4:51-61, 4:67-5:17; Fig. 4. Jenett's block allocation map is "a physical to virtual map which associates particular physical sectors of the flash medium with a related virtual address." Ex. 1033, 6:16-19; Baker 181 (explaining that "virtual address" refers to a logical address). These mappings are maintained by the flash controller. Ex. 1025, 1:65-67, 2:1-8, 3:56-4:5, 4:56-61. Thus, Jenett's FTL is managed by the flash controller and "maps logical block addresses to corresponding respective physical block addresses of the flash memory device," as recited in the claim. Baker ¶¶ 245-246; Ex. 1025, 2:37-40.

Thus, it would have been obvious that the SSD in the Shu Patent would have used Jenett's FTL as maintained by Ban's flash controller, and this claim element, as interpreted by UTL, would have been obvious to a POSITA. Baker ¶¶ 246-247.

e) Element 12[e].

UTL accuses the Trim command of infringing this element. Ex. 1013 at 7. But because the ’727 Patent lacks priority to the 2006 Provisional as explained in Section VI, above, the accused command is prior art, regardless of how UTL construes this term.

The remove-on-delete/Trim command specifies a logical block address. Ex. 1003, 3:17-20 (“this includes … logical block addressing (‘LBA’) typically followed by the length”); Ex. 1002, [0014]; Ex. 1017 § 3 (specifying LBA and Count); Ex. 1018 § 6.1.1. The remove-on-delete/Trim command is received by the SSD through the interface 120 shown in figure 1. Ex. 1003, 2:35-51, 3:20-22, 4:4-29, Fig. 1; Ex. 1002, [0012], [0014], [0016]-[0019], Fig. 1. Section IX.G.c (element 12[c]), above, explains that the storage processor receive the commands from the interface. For the reasons discussed in section XI.A.f (element 1[f]) above, it would have been obvious to receive the Trim command to indicate invalid data, in place of sending the file indication map as taught by Jenett, because the Trim command more efficiently specifies a logical address.

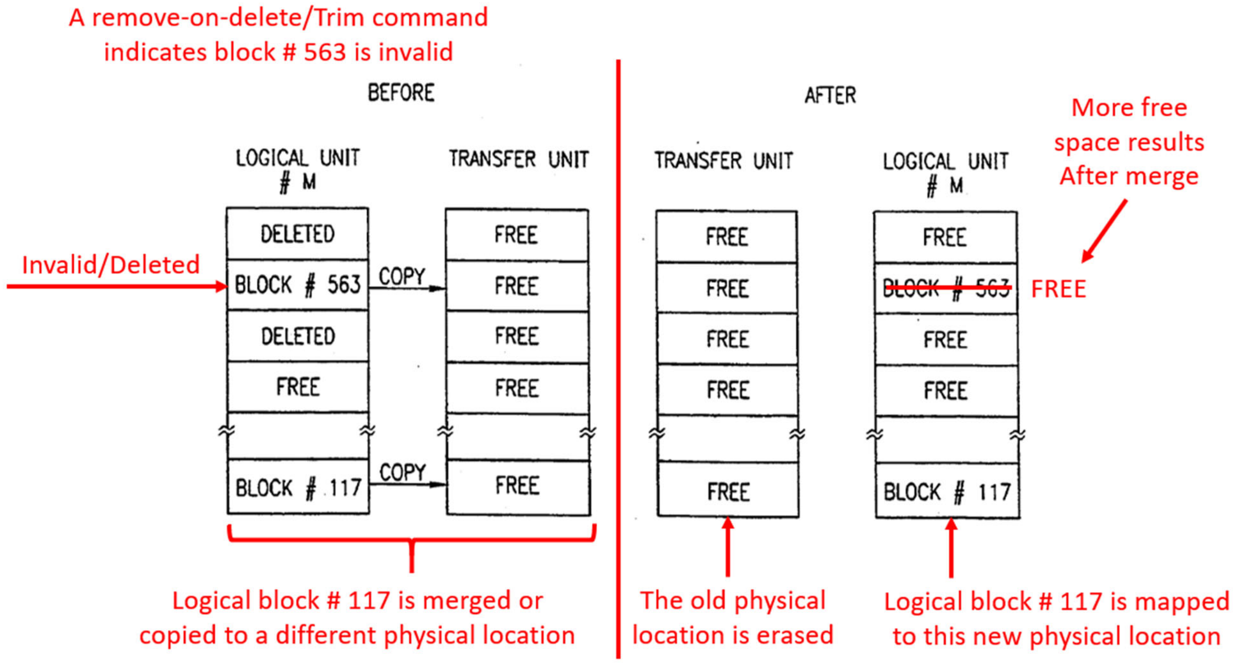
Thus, this element would have been obvious to a POSITA when applying UTL’s interpretations. Baker ¶¶ 249-250.

f) Element 12[f].

UTL accuses the FTL of infringing this element. Ex. 1013 at 7. The industry credits Ban for teaching the FTL. Baker ¶ 242; Ex. 1033, 5:21-26.

The remove-on-delete/Trim command "includes the invalid data information and instructs the SSD device … to mark the indicated data as invalid." Ex. 1003, 4:65-5:1; Ex. 1002, [0023]. "Invalid or removed data is thus removed from consideration by an SSD device as valid data. This frees the SSD device from performing any operations to preserve or maintain such data." Ex. 1003, 4:38-41; Ex. 1002, [0020].

As discussed in Section IX.G.d (element 12[d]), above, it would have been obvious that Shu's SSD uses an FTL such as taught by Ban and incorporated in Jenett. This FTL indicates the status of blocks, including whether blocks are "valid" or "other than valid." Ex. 1033, 6:18-23. Given the Shu Patent's instruction to mark data invalid, a POSITA would have found it obvious for the FTL to mark the blocks as invalid. Baker ¶¶ 253-254. Then the blocks would not be preserved during a merge operation, such as explained in Section IX.A.f (element 1[f]), above. The resulting effect is shown below.


A remove-on-delete/Trim command indicates block # 563 is invalid
BEFORE
LOGICAL UNIT # M
TRANSFER UNIT
DELETED
Invalid/Deleted
BLOCK # 563
COPY
DELETED
FREE
BLOCK # 117
COPY
FREE
AFTER
TRANSFER UNIT
LOGICAL UNIT # M
More free space results After merge
~~BLOCK # 563~~
FREE
BLOCK # 117
Logical block # 117 is merged or copied to a different physical location
The old physical location is erased
Logical block # 117 is mapped to this new physical location

Thus, when applying UTL's interpretation, a POSITA would have found this element obvious. Baker ¶¶ 252-256.

g) Element 12[g].

As best understood, UTL accuses the Trim command of indicating that data corresponding to the LBA in the Trim command is deleted at the storage client. Ex. 1013 at 7. But because the '727 Patent lacks priority to 2006, the accused command is prior art, regardless of how UTL construes this term.

As discussed in Section IX.G.f (element 12[f]), above, the FTL responds to the remove-on-delete/Trim command by indicating that data is invalid. Ban and Jenett further teach that these status indicators for each block are stored as persistent data in flash memory because they are stored in a Block Allocation Map ("BAM") that resides in persistent flash memory. Ex. 1025, 4:36-40, 4:45-50, 7:51-52, Fig.

3. Jenett, which uses Ban's FTL technology, also shows that the BAM is stored in the flash memory card. Ex. 1033, Fig. 4. Thus, a POSITA would have understood that a status indicator would be stored as persistent data on the flash memory device in response to the remove-on-delete/Trim command, so the claim would have been obvious to a POSITA when applying UTL's interpretation. Baker ¶¶ 258-259.

H. Claim 13.

Section IX.G.d (element 12[d]), above, explains how a POSITA would have found it obvious that the Shu Patent's SSD would have used Jenett's FTL. Jenett shows that the FTL structures are stored in the "flash memory card." Ex. 1033, Fig. 4. For the reasons discussed in Section IX.G.g (element 12[g]), above, the BAM and maps of the FTL are stored in the flash memory device. Thus, a POSITA would have found this claim obvious. Baker ¶ 261.

I. Claim 14.

Section IX.G.c (element 12[c]), above, explains how a POSITA would have found it obvious that the Shu Patent's SSD would have generic components including RAM (the claimed "volatile memory device") coupled to the flash controller (the claimed "storage processor").

A POSITA would have found it obvious to store part of Ban's FTL in the form of a secondary map in RAM for faster access by applying a concept known as "caching." *Id.* ¶ 264. Ban applies this concept when teaching that a "secondary virtual map residing in volatile random access memory can be reconstructed by scanning, at startup, the block usage map that resides at the top of each unit. Blocks marked as mapped to a virtual address are identified, and the secondary virtual map is constructed accordingly." Ex. 1025, 7:53-58. In some cases, this secondary map is a small part of the primary map, in accordance with caching principles. Ex. 1025, 3:3-6. Changes to the virtual map in flash memory are reflected in the secondary map in RAM. *Id.*, 3:17-20.

Thus, a POSITA would have found this claim obvious. Baker ¶¶ 263-265.

J. Claim 15.

UTL accuses an SSD of infringing this element for responding to read requests by returning zeros, as in the ATA standard. Ex. 1013 at 8. Frank Shu proposed this in the Shu Trim Proposals, revision 1. Ex. 1018. The section "Deallocate(Trim)" 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." Ex. 1018 at 7. Thus, when applying the UTL's interpretation, this claim would have been obvious to a POSITA because Frank Shu proposed it for the ATA industry standard. Baker ¶ 267-269.

K. Claim 16.

When applying the UTL's interpretation, this claim would have been obvious to a POSITA for the same reasons discussed in section XI.J (claim 15) above. Baker

¶¶ 271-273; Ex. 1018 at 7 (“If a read occurs to any part of the data set before it is written, the device … may return all 0s.”).

XII. Ground 2: Obvious Over the Shu Patent, the Shu Trim Proposals, Ban, and Further in View of Jenett.

Claims 1-6 and 12-16 are rendered obvious by the Shu Patent in view of the Shu Trim Proposal revisions 0–1, Ban, and Jenett for the same reasons discussed in Ground 1. If the Board finds that Jenett does not incorporate the relied-upon sections of Ban by reference, then Ground 2 differs from Ground 1 solely by combining Jenett and Ban under § 103 instead of incorporating by reference. A POSITA would have been motivated to combine Jenett and Ban for the same reasons explained in Ground 1. Baker ¶ 274.

XIII. 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 Proposal at least by April 21, 2007, months before the earliest possible September 2007 priority date. *See supra* Section VI. 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). Thus, Exhibits 1002-1003, 1017-1018, 1028, and 1029 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.

XIV. 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 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 XIV, has been served with a complaint alleging infringement of the patent at issue herein.

B. Related Proceedings

UTL asserted the '727 Patent against Petitioners in the Western District of Texas, Case No. 6:20-cv-500. In the same Court, UTL also asserted the '727 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. 8,533,406), IPR2021-00344 (Pat. 8,762,658), and IPR2021-00345 (Pat. 9,632,727). Petitioners are also filing contemporaneously two additional IPR proceedings that challenge the priority date of the same patent family: IPR2021-00343 (8,533,406) and IPR2021-00344 (Pat. 8,762,658).

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: 650.858.6500, 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: 650.858.6500, 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

XV. Fees

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

XVI. 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: 650.858.6500, 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: 650.858.6500, 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.; and Micron Technology Texas LLC

CLAIM LISTING

Claim 1	
Element	**Language**
1[a]	An apparatus, comprising:
1[b]	a solid-state storage medium;
1[c]	a solid-state storage controller configured to implement storage operations on the solid state storage medium in response to requests from a computer system,
1[d]	including storing data pertaining to logical addresses of a logical address space at respective physical addresses of the solid-state storage medium; and
1[e]	an indexer, comprised within the solid-state storage controller, wherein the indexer is configured to assign logical addresses of the logical address space to physical addresses in use to store data pertaining to the logical addresses on the solid-state storage medium;
1[f]	wherein the indexer is further configured to remove an assignment between an identified logical address and a physical address of the solid-state storage medium in response to a message received from a host operating system,
1[g]	the message indicating that the identified logical address is erased.

Claim 2	
Element	**Language**
2[a]	The apparatus of claim 1, wherein the indexer assigns logical addresses to physical addresses by use of index entries, and

2[b]	wherein the indexer removes an index entry corresponding to the identified logical address in response to the message.

Claim 3	
Element	**Language**
3	The apparatus of claim 1, wherein the indexer assigns logical addresses to physical addresses by use of index metadata maintained in a memory of the storage controller.

Claim 4	
Element	**Language**
4	The apparatus of claim 1, wherein the indexer comprises firmware of the solid-state storage controller.

Claim 5	
Element	**Language**
5	The apparatus of claim 1, further comprising a garbage collector configured to designate that the physical address previously assigned to the identified logical address comprises data suitable for removal from the solid-state storage medium in response to the message.

Claim 6	
Element	**Language**
6	The apparatus of claim 1, wherein the solid-state storage controller comprises a bus interface configured to communicatively couple the solid-state storage controller to the computer system, wherein the bus interface comprises one of … an AT Attachment (ATA) interface, a Parallel ATA (PATA)

	interface, a Serial ATA (SATA) bus interface, an external SATA bus interface …

Claim 12

Element	**Language**
12[a]	A non-volatile solid-state storage system, comprising:
12[b]	a storage interface configured to communicate with a storage client;
12[c]	a storage processor coupled to the storage interface; a flash memory device coupled to the storage processor; and
12[d]	a logical-to-physical translation layer maintained by the storage processor, wherein the logical-to-physical translation layer maps logical block addresses to corresponding respective physical block addresses of the flash memory device,
12[e]	wherein the storage processor is configured to: receive, from the storage client through the storage interface, an empty-block directive command and a range of logical block addresses,
12[f]	wherein the storage processor is configured to: … update the logical-to-physical translation layer to indicate that data stored in physical block addresses corresponding to the received logical block addresses do not need to be preserved, and
12[g]	store persistent data on the flash memory device, the persistent data indicating that the data corresponding to the received logical block addresses is deleted at the storage client.

Claim 13

Element	**Language**
13	The system of claim 12, wherein the logical-to-physical translation layer is stored in the flash memory device.

Claim 14	
Element	**Language**
14	The system of claim 12, further comprising a volatile memory device coupled to the storage processor, wherein the logical-to-physical translation layer is stored in the volatile memory device.

Claim 15	
Element	**Language**
15	The system of claim 12 wherein the storage process[o]r is configured such that, responsive to receiving a read request specifying one or more logical addresses included in the empty-block directive command, the storage processor returns a predetermined data string.

Claim 16	
Element	**Language**
16	The system of claim 15, wherein data bits of the predetermined data string have a uniform logic level.

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 12,896 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 was 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: 650.858.6500, 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 foregoing "**PETITION FOR *INTER PARTES* REVIEW OF CLAIMS 1-6 AND 12-16 OF U.S. PATENT NO. 9,632,727**, " Petitioners' Power of Attorney and Exhibits 1001 – 1040 by FedEx on the Patent Owner at the correspondence address of record for U.S. Patent No. 9,632,727:

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 Patent Owner via email:

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

WINSTON & STRAWN LLP

/s/ *Katherine A. Vidal*
Katherine A. Vidal
Winston & Strawn LLP
275 Middlefield Rd, Suite 205
Menlo Park, California 94025
kvidal@winston.com
T: 650.858.6500, F: 650.858.6550
USPTO Reg. No. 46,333