

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICRON TECHNOLOGY, INC., MICRON SEMICONDUCTOR
PRODUCTS, INC., MICRON TECHNOLOGY TEXAS LLC,
DELL TECHNOLOGIES INC., DELL INC., and HP INC.,
Petitioner,

v.

UNIFICATION TECHNOLOGIES LLC,
Patent Owner.

IPR2021-00345
Patent 9,632,727 B2

Before JUSTIN T. ARBES, TERRENCE W. McMILLIN, and
CHRISTOPHER L. OGDEN, *Administrative Patent Judges*.¹

OGDEN, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

¹ Katherine K. Vidal, Under Secretary of Commerce for Intellectual Property and Director of the United States Patent and Trademark Office, is recused from this proceeding and took no part in this decision. *See* Director's Memorandum, Procedures for Recusal to Avoid Conflicts of Interest and Delegations of Authority (Apr. 20, 2022), <https://go.usa.gov/xJjch>; Interim Process for Director Review § 20, <https://go.usa.gov/xJjce>.

I. INTRODUCTION

In response to a Petition (Paper 4, “Pet.”) filed by Petitioners Micron Technology, Inc., Micron Semiconductor Products, Inc., Micron Technology Texas LLC, Dell Technologies Inc., Dell Inc., and HP Inc. (collectively, “Petitioner”), the Board instituted an *inter partes* review of claims 1–6 and 12–16 of U.S. Patent No. 9,632,727 B2 (Ex. 1001, “the ’727 patent”). (Paper 9, “Dec.”). Patent Owner Unification Technologies LLC (“UTL”) filed a Patent Owner Response (Paper 21, “PO Resp.”), Petitioner filed a Reply to the Patent Owner Response (Paper 28, “Pet. Reply”), and UTL filed a Sur-reply (Paper 33, “PO Sur-reply”).

We held an oral hearing on April 13, 2022, and the transcript is entered on the record. Paper 36 (“Tr.”).

This is a final written decision under 35 U.S.C. § 318(a) as to whether the claims challenged in the *inter partes* review are unpatentable. For the reasons below, we conclude that Petitioner has shown that all the challenged claims are unpatentable on at least one ground of the Petition.

II. BACKGROUND

A. RELATED PROCEEDINGS

The parties identify the following as related matters: *Unification Technologies LLC v. Dell Technologies, Inc.*, No. 6:20-cv-499-ADA (W.D. Tex. filed June 5, 2020), *Unification Technologies LLC v. Micron Technology, Inc.*, No. 6:20-cv-500-ADA (W.D. Tex. filed June 5, 2020) (“the district court case”), and *Unification Technologies LLC v. HP Inc.*, No. 6:20-cv-501-ADA (W.D. Tex. filed June 5, 2020). Pet. 67; Paper 6, 2–3.

Petitioner also filed petitions challenging claims of patents related to the '727 patent in IPR2021-00343 and IPR2021-00344, for which the Board issued final written decisions on July 8, 2022.

B. THE '727 PATENT (EX. 1001)

The '727 patent discloses a technique “for managing a non-volatile storage medium.” Ex. 1001, code (57). According to the patent, at the time of the claimed invention there were at least two known strategies for a file system to delete data in non-volatile storage media when that data is no longer useful. First, it could issue an erase command that “deletes a directory entry in the file system while leaving the data in place in the storage device containing the data.” *Id.* at 1:32–35. “Typically, a data storage device is not involved in this type of erase operation.” *Id.* at 1:35–36. Thus, according to UTL, this approach has the drawback of “the data storage device being unaware [when] data has become invalid.” PO Resp. 2 (citing Ex. 1001, 1:35–36).

In the second strategy, the file system could “write zeros, ones, or some other null data character to the data storage device to actually replace the erased file.” Ex. 1001, 1:37–39. But according to the '727 patent, “this is inefficient because valuable bandwidth is used while transmitting” the highly-redundant overwriting data. *Id.* at 1:39–41, 1:51–60. Also, the '727 patent states that the approach is ineffective when used in solid-state storage devices, because such devices typically do not have the ability to overwrite previously stored data to erase it. *Id.* at 1:43–50. For example, flash memory is a solid-state storage device that stores data in “blocks,” each block containing smaller “pages” of data. *Id.* at 6:66–7:1, 17:30–33. Such devices

cannot write a page of new data over a page of old data without first erasing the entire block containing the old data. *See* Pet. 3 (citing Ex. 1004 ¶ 73); PO Resp. 13 (citing Ex. 1003, 4:37–38).

The '727 patent attempts to overcome these drawbacks. *See* Ex. 1001, 1:64–2:6. In the claimed invention, data are represented by two types of addresses: physical addresses that indicate the physical location of the data on the storage medium, and logical addresses that identify data logically while internally mapping the data to associated physical addresses. *See id.* at 2:32–34. When a computer system deletes data corresponding to a particular logical address, it sends a message to a storage controller for the medium. *Id.*, code (57). This “message may comprise a hint, directive, or other indication that the data [associated with the logical address] has been erased and/or deleted.” *Id.* In response to this message, “the storage controller records an indication that the contents of a . . . physical address associated with the logical [address] do not need to be preserved on the non-volatile storage medium.” *Id.*

Figure 1A of the '727 patent, which we reproduce below, is a schematic block diagram of the claimed invention. Ex. 1001, 4:35–38, 6:56–59.

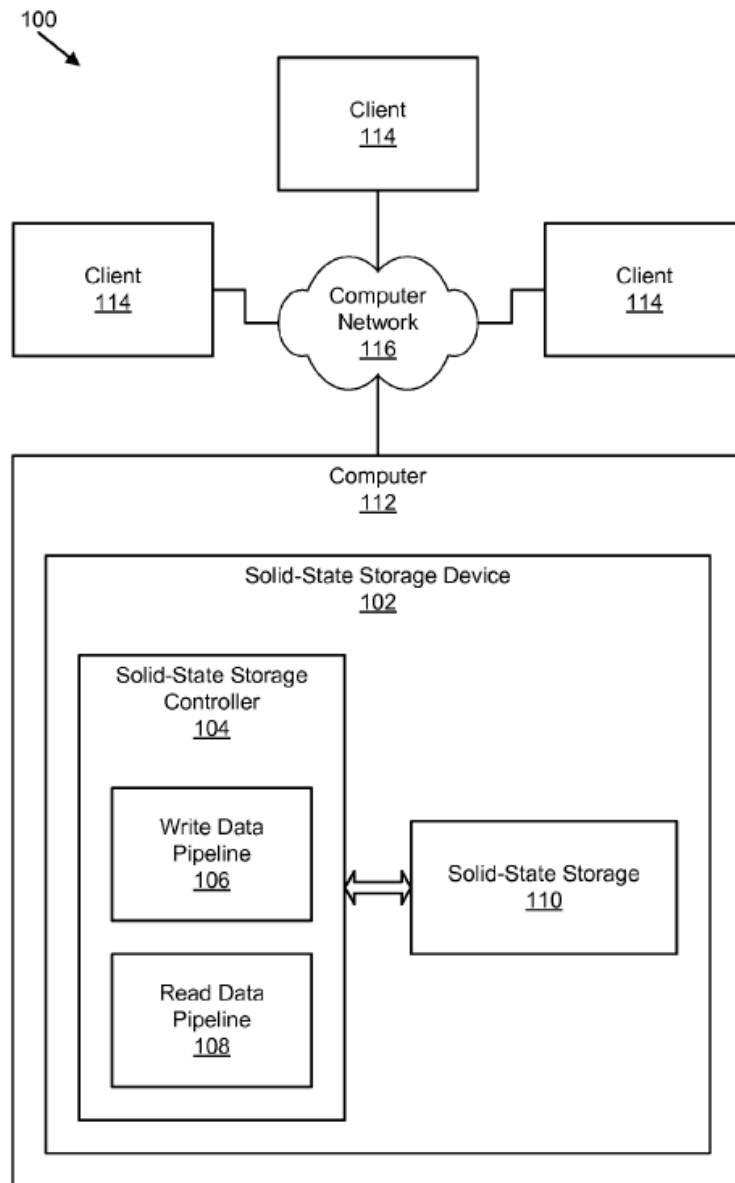


FIG. 1A

As shown above, overall system 100 includes clients 114, which communicate over computer network 116 with computer 112 having solid-state storage device 102. Ex. 1001, 6:56–63, 7:5:7. Solid-state storage device 102 includes one or more solid-state storage devices 110 (e.g., flash

memory) and one or more solid-state storage controllers 104,² which write to storage devices 110 through write data pipeline 106 and read from storage devices 110 through read data pipeline 108. *Id.* at 6:59–7:4, 7:34–37.

Storage controller 104 includes an “object index module.” Ex. 1001, 10:50–51, Fig. 2A (item 266). When a “data packet is stored and the physical address of the data packet is assigned” to a data object, this object index module creates an entry in an index that maps the object’s “logical identifier” to “one or more physical addresses corresponding to where the storage controller [104 has] stored . . . data packets and any object metadata packets.” *Id.* at 12:1–9. When storage controller 104 receives “a command that erases a data block or other object elements,” storage controller 104 stores an “erase packet that includes . . . a reference to the object, [a] relationship to the object, . . . the size of the data block erased,” and optionally an “indicat[ion] that the erased object elements are filled with zeros.” *Id.* at 12:25–32. “Thus, the erase [command] can be used to emulate actual memory or storage that is erased and actually has a portion of the appropriate memory/storage actually stored with zeros in the cells of the memory/storage.” *Id.* at 12:33–36.

Write data pipeline 106 includes a garbage collection system with a garbage collector bypass that “receives data segments from the read data pipeline 108.” Ex. 1001, 27:45–48, Fig. 3. This garbage collection system “marks packets that are no longer valid, typically because the packet is marked for deletion or has been modified and the modified data is stored in a

² The ’727 patent also refers to a particular storage controller as “storage controller 152.” *See, e.g.*, Ex. 1001, 8:29. For clarity and consistency, we use Figure 1A’s identification of a storage controller as item 104.

different location.” *Id.* at 27:49–52. When a particular section of storage needs to be recovered, such as when the section contains a large number of invalid packets, the garbage collection bypass can relocate any still-valid data in that section. *Id.* at 27:62–63. This relocation proceeds by “allow[ing] packets to be read into the read data pipeline 108 and then transferred directly to the write data pipeline 106 without being routed out of the solid-state storage controller 104.” *Id.* at 27:63–67.

C. CHALLENGED CLAIMS AND GROUNDS

Independent claim 1, which exemplifies the other challenged claims of the ’727 patent, is as follows:

- 1[a] 1. 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, including storing data pertaining to logical addresses of a logical address space at respective physical addresses of the solid-state storage medium; and
- 1[d] 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[e] 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, the message indicating that the identified logical address is erased.

Ex. 1001, 53:22–40 (formatting and reference letters added); Pet. 70–71 (claim listing). Dependent claims 2–6, which Petitioner also challenges, depend directly from claim 1. *See id.* at 53:41–54:3.

Petitioner also challenges independent claim 12, which is as follows:

- 12[a] 12. 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;
- 12[d] a flash memory device coupled to the storage processor; and
- 12[e] 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, wherein the storage processor is configured to:
 - 12[f] receive, from the storage client through the storage interface, an empty-block directive command and a range of logical block addresses,
 - 12[g] 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[h] 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.

Ex. 1001, 54:42–64 (formatting and reference letters added); Pet. 73–75 (claim listing). Dependent claims 13–16, which Petitioner also challenges, depend directly or indirectly from claim 12. *See id.* at 54:65–55:11.

Petitioner argues three grounds for *inter partes* review, as summarized in the following table:

Ground	Claims Challenged	35 U.S.C. §	Reference(s)/Basis
1	1–6, 12–16	103(a) ³	Bennett ⁴
2	1–3, 5, 6, 12–16	103(a)	Suda ⁵
3	4, 13	103(a)	Suda, Bennett

Pet. 9–10. For each of the grounds, Petitioner also relies on the general knowledge of a person of ordinary skill in the art. Pet. 9–10 (citing *Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337–38 (Fed. Cir. 2020)).

D. DECLARATORY TESTIMONY

Petitioner submits a declaration from Jacob Baker, Ph.D., P.E. as expert testimony. Ex. 1004 ¶ 1; *see also* Ex. 1026 (curriculum vitae). Exhibit 2012 is the transcript for his deposition, dated September 16, 2021.

UTL submits a declaration from Vijay K. Madiseti, Ph.D. Ex. 2010; *see also* Ex. 2011 (curriculum vitae). Exhibit 1038 is the transcript for his deposition, dated January 7, 2022.

³ 35 U.S.C. § 103(a) (2006), *amended by* Leahy–Smith America Invents Act, Pub. L. No. 112-29 § 103, sec. (n)(1), 125 Stat. 284, 287, 293 (2011) (effective Mar. 16, 2013). The ’727 patent claims priority to a provisional application filed on December 6, 2006, which is before the effective date of this amendment to § 103. *See* Ex. 1001, code (60); Pet. 4 (“Solely for purposes of this IPR, Petitioners assume, but do not concede, an effective filing date of December 6, 2006, for the ’727 patent.”).

⁴ Bennett et al., US 7,624,239 B2, filed Nov. 14, 2005, issued Nov. 24, 2009 (Ex. 1002). Petitioner alleges that Bennett is prior art under 35 U.S.C. § 102(e). Pet. 10. UTL does not contest this in its Response.

⁵ Suda et al., US 7,057,942 B2, published Mar. 16, 2006, issued June 6, 2006 (Ex. 1003). Petitioner alleges that Suda is prior art under 35 U.S.C. § 102(a). Pet. 10. UTL does not contest this in its Response.

III. GROUNDS OF THE PETITION

For the reasons below, we determine that Petitioner has shown, by a preponderance of the evidence, that claims 1–6 and 12–16 of the '727 patent are unpatentable under the grounds based on Suda (for claims 1–3, 5, 6, 12, and 14–16) or the combination of Suda and Bennett (for claims 4 and 13). We do not reach Petitioner's ground relying on Bennett alone, and we do not reach Petitioner's alternative argument for claim 13 relying on Suda alone. Before analyzing these grounds in detail, we address two matters that will underlie our analysis: the level of ordinary skill in the art and the construction we will apply to the claim terms.

A. LEVEL OF ORDINARY SKILL IN THE ART

The level of ordinary skill in the pertinent art at the time of the invention is a factor in how we construe patent claims. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc). It is also one of the factors we consider when determining whether a patent claim is obvious over the prior art. *See Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

To assess the level of ordinary skill, we construct a hypothetical “person of ordinary skill in the art,” from whose vantage point we assess obviousness and claim interpretation. *See In re Rouffet*, 149 F.3d 1350, 1357 (Fed. Cir. 1998). This legal construct “presumes that all prior art references in the field of the invention are available to this hypothetical skilled artisan.” *Id.* (citing *In re Carlson*, 983 F.2d 1032, 1038 (Fed. Cir. 1993)).

Petitioner argues that a person of ordinary skill would have had “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.” Pet. 4.

Professor Baker also opines that a person of ordinary skill would have had certain background knowledge about flash memory management, including

how flash memory erases data, how flash memory programs or writes data, how memory is used in a cache hierarchy, relative speeds of flash memory compared to other memory, how garbage collection is used with flash memory, how to use wear leveling to combat endurance limits of flash memory, how the [Flash Translation Layer (“FTL”)] works, and industry standards affecting flash memory including the [Advance Technology Attachment (“ATA”)] standard.

Pet. 5 (citing Ex. 1004 ¶ 61); *see also* Ex. 1004 ¶¶ 57, 62–95 (describing flash memory technologies that, according to Professor Baker, were well understood and widely used before the priority date of the ’727 patent).

For this proceeding, UTL does not contest Petitioner’s proposed articulation of the level of ordinary skill, and Dr. Madisetti applied this articulation in his analysis. *See* PO Resp. 16; Ex. 2010 ¶¶ 47–52. Because Petitioner’s uncontroverted articulation of the level of ordinary skill in the art is supported by testimonial evidence and appears consistent with the types of problems and solutions in the ’727 patent, we adopt it for this decision. *See, e.g.*, Ex. 1001, 1:25–60 (“Background of the Invention” section of the ’727 patent, describing known write, read, and erase procedures for solid-state storage devices).

B. CLAIM CONSTRUCTION

In an *inter partes* review, we construe a patent claim “using the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b).” 37 C.F.R. § 42.100(b) (2021). This generally includes “construing the claim in accordance with the ordinary and

customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” *Id.* The ordinary and customary meaning of a claim term “is its meaning to the ordinary artisan after reading the entire patent,” and “as of the effective filing date of the patent application.” *Phillips*, 415 F.3d at 1313, 1321. There are only two circumstances in which a construction departs from the ordinary and customary meaning: “1) when a patentee sets out a definition and acts as [their] own lexicographer, or 2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution.” *Thorner v. Sony Comput. Entm’t Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012). Any such special meaning of a term “must be sufficiently clear in the specification that any departure from common usage would be so understood by a person of experience in the field of the invention.” *Multiform Desiccants Inc. v. Medzam Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998).

To construe the claim terms, “we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006).

Petitioner provides a listing of the parties’ proposed claim constructions in the district court case, but argues that “[t]hese construction disputes do not affect the outcome of [the] Petition with respect to any claim.” Pet. 5–7. Although Petitioner asserted in district court that certain terms are indefinite, Petitioner states that it has applied UTL’s proposed constructions of those terms in its Petition. *Id.* at 7–8 (citing *Spherix Inc. v. Matal*, 703 F. App’x 982, 983 (Fed. Cir. 2017); *Target Corp. v. Proxicom Wireless, LLC*, IPR2020-00904, Paper 11 at 12 (PTAB Nov. 10, 2020);

Samsung Elecs. Am., Inc. v. Prisia Eng'g Corp., 948 F.3d 1342, 1355 (Fed. Cir. 2020); *Intel Corp. v. Alacritech, Inc.*, IPR2017-01391, Paper 8 at 7 (PTAB Nov. 28, 2017); *Vibrant Media v. Gen. Elec. Co.*, No. IPR2013-00172, Paper 50, 10 (PTAB July 28, 2014)).

After Petitioner filed the Petition, the district court issued a Claim Construction Order interpreting five terms of the '727 patent, as shown in the following table:

Claim Term	District Court's Construction
<i>empty-block directive command</i> (claims 12, 15)	"A command that indicates that certain blocks contain data that does not need to be preserved"
<i>the identified logical address is erased</i> (claim 1)	"Plain and ordinary meaning"
<i>logical-to-physical translation layer</i> (claim 12)	"Not indefinite; not subject to § 112(f); plain and ordinary meaning"
<i>indexer</i> (claims 1–4)	"Not indefinite; not subject to § 112(f); plain and ordinary meaning"
<i>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 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</i> (claim 12)	"Not indefinite; not subject to § 112(f); plain and ordinary meaning"

Ex. 2006, 2–3; *accord* PO Resp. 16–17.

The parties accept the district court's constructions, except that the parties have different views on the meaning of *the identified logical address*

is erased. See PO Resp. 17–18; Pet. Reply 10–11. In district court, Petitioner argued that this term has its “[p]lain and ordinary meaning,” which “is that the identified logical address is erased, not the data associated with the identified logical address is erased.” Pet. 6; Ex. 1032, 13; Ex. 1033, 2. UTL argued that the term means “the data identified by the logical address does not need to be preserved.” Pet. 6; Ex. 2007, 11–13. The district court interpreted the term to have its “[p]lain and ordinary meaning,” but otherwise, “the Court did not believe that the term needed to be construed.” Ex. 2014, 1 (clarifying email from the district court’s law clerk); *see also* Ex. 2006, 2.

In our Decision instituting trial, we agreed with Petitioner and with the district court that the term has its ordinary and customary meaning because neither party had pointed to any evidence “that the ’727 patent clearly defines this term in the specification or that the applicant has otherwise clearly disclaimed any of the term’s ordinary and customary scope.” Dec. 14. Likewise, neither party presented such evidence at trial, and UTL did not argue that the term departs from its ordinary and customary meaning. See PO Resp. 17–20. Thus, given the evidence of record, we agree with the district court that the term *the identified logical address is erased* has its ordinary and customary meaning. However, it remains for us to determine the scope of that meaning sufficiently to resolve the parties’ controversy.

At trial, UTL argues that we should construe *the identified logical address is erased* to mean “data associated with the identified logical address is erased.” PO Resp. 17–18 (citing Ex. 2010 ¶¶ 53–56). According to UTL, “[t]his construction aligns with industry usage” and “aligns with the

specification, which recognizes that file systems can erase data associated with a logical address without involving the storage device.” *Id.* at 18 (citing Ex. 1001, 1:32–36; Ex. 2007, 2017–2034). UTL also argues that UTL’s proposed construction finds support in Petitioner’s own contentions, which UTL characterizes as arguing “that removing ‘information about which logical identifiers are associated with [which photos]’ satisfies the requirement that ‘the identified logical address is erased.’” *Id.* at 19 (quoting Pet. 45).

But in its Reply, Petitioner disagrees that the term *logical address* includes associated data. Pet. Reply 10–11. According to Petitioner, “claim 1 separately recites both ‘logical address’ and ‘data,’ for example, ‘storing data pertaining to logical addresses,’” so construing logical addresses to include associated data would render the claim unintelligible. *Id.* (quoting Ex. 1001, 53:27). Petitioner also argues that the ’727 patent provides examples of deleting a logical identifier without deleting associated data. *Id.* at 11 (citing Ex. 1001, 2:25–30, 3:11–15).

We agree with Petitioner that the plain language of claim 1 distinguishes between a “logical address” and “data pertaining to” the logical address which is stored at a physical address on the solid-state storage medium. *See* Ex. 1001, 53:27–29, 53:33–34. We also agree with Petitioner that in the ’727 patent disclosure, “deleting an index entry that associates the logical identifier with the physical storage location” or “deleting a mapping between the logical identifier and the physical storage location” would correspond to erasing the “identified logical address” as recited in claim 1. *See* Ex. 1001, 2:25–28, 53:39–40. In these examples, the ’727 patent does not disclose that the data associated with the logical address

is deleted, just that the link between the logical identifier and the physical data has been deleted.

Thus, we disagree with UTL’s proposed construction of *the identified logical address is erased* to mean “data associated with the identified logical address is erased.” Other than this, we need not further construe the term, or explicitly construe any other terms, to resolve issues raised in the *inter partes* review. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

C. OBVIOUSNESS GROUNDS BASED ON SUDA (CLAIMS 1–3, 5, 6, AND 12–16)

Turning to the grounds of the Petition, we begin with the second ground, which asserts that claims 1–3, 5, 6, and 12–16 are unpatentable under 35 U.S.C. § 103(a) as obvious over Suda. *See* Pet. 9–10, 39–60.

A claim is unpatentable under § 103(a) for obviousness if the differences between the claimed subject matter and the prior art are “such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). When a ground in a petition is based on a combination of references, we consider “whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *Id.* at 418 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

We base our obviousness inquiry on factual considerations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) any objective indicia of obviousness or non-obviousness that may be in evidence. *See Graham*, 383 U.S. at 17–18.

Considering these factors,⁶ we determine that Petitioner has shown, by a preponderance of the evidence, that claims 1–3, 5, 6, and 12–16 are unpatentable under § 103(a) as obvious over Suda. We begin our analysis with a brief overview of Suda, and then we address the parties’ contentions with respect to the challenged claims.

1. Suda

Suda discloses a “memory management device for managing a nonvolatile semiconductor memory.” Ex. 1003, code (57). Figure 1 of Suda, reproduced below, is a block diagram illustrating this device. Ex. 1003, 2:58–60.

⁶ Petitioner presents evidence of simultaneous invention as an objective indicium of obviousness. Pet. 62–63. UTL does not address this in its Response, and does not otherwise present evidence of objective indicia of non-obviousness. Because Petitioner has shown obviousness based on the other *Graham* factors, and any objective indicia of obviousness could only further support Petitioner’s arguments, we need not consider this evidence in our decision.

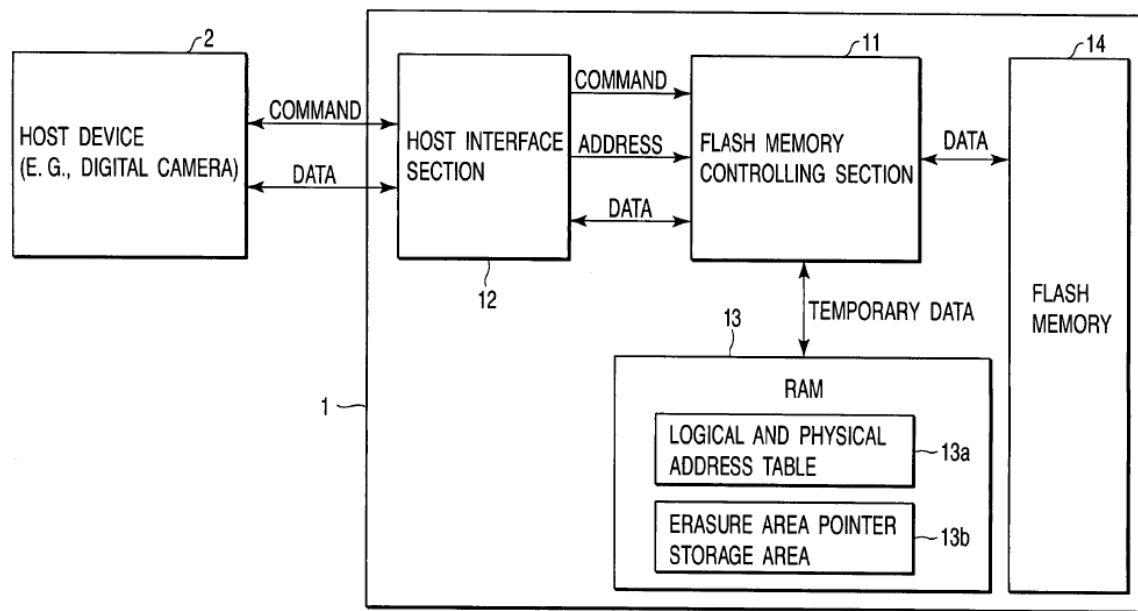


FIG. 1

As shown above, Figure 1 depicts host device 2, which may be a digital camera, in communication with memory card 1 comprising host interface section 12, random access memory (RAM) 13, and flash memory 14 (e.g., a NAND type nonvolatile memory), each of which is connected to flash memory controlling section 11. *Id.* at 2:61–3:3.

“[F]lash memory controlling section 11 manages data erasure and a table indicating a relationship between logical blocks and physical blocks of the flash memory 14.” Ex. 1003, 3:13–15. Logical and physical address table 13a, stored in RAM 13, “manages logical addresses and physical addresses allocated to physical blocks in which data items are written, of the physical blocks in the flash memory 14, in association with each other.” *Id.* at 3:41–47.

According to Suda, when memory controlling section 11 erases a subset of pages of a physical block, “the time required for data erasure is

long” because the non-erased pages must be read and “written to another physical block.” Ex. 1003, 4:60–67; *see also id.* at 1:19–23. Suda discloses an improved process that instead writes “erasure area pointer[s]” to erasure area pointer storage area 13b in RAM 13. *Id.* at 5:14–16. These pointers indicate that pages in a particular address range are in a “virtual erased state . . . in which the flash memory controlling section 11 does not actually erase data items to be erased, i.e., they are subjected to virtual erasure, in response to an access command from the host device 2.” *Id.* at 5:19–27.

Specifically, when host device 2 issues a data read command with a logical block address, flash memory controlling section 11 obtains the corresponding physical block address from logical and physical address table 13a, then looks to erasure area pointer storage area 13b to determine whether the requested data is within “an area indicated by the erasure area pointer or pointers”; if so, flash memory controlling section 11 outputs “initial-value” (i.e., empty) data. Ex. 1003, 9:52–62. As a result, “it is not necessary to transfer data . . . and the time for erasing data can be greatly shortened.” *Id.* at 9:63–65.

Figure 8 of Suda, reproduced below, is a flowchart depicting the process for erasing data on the memory device. Ex. 1003, 2:38–40.

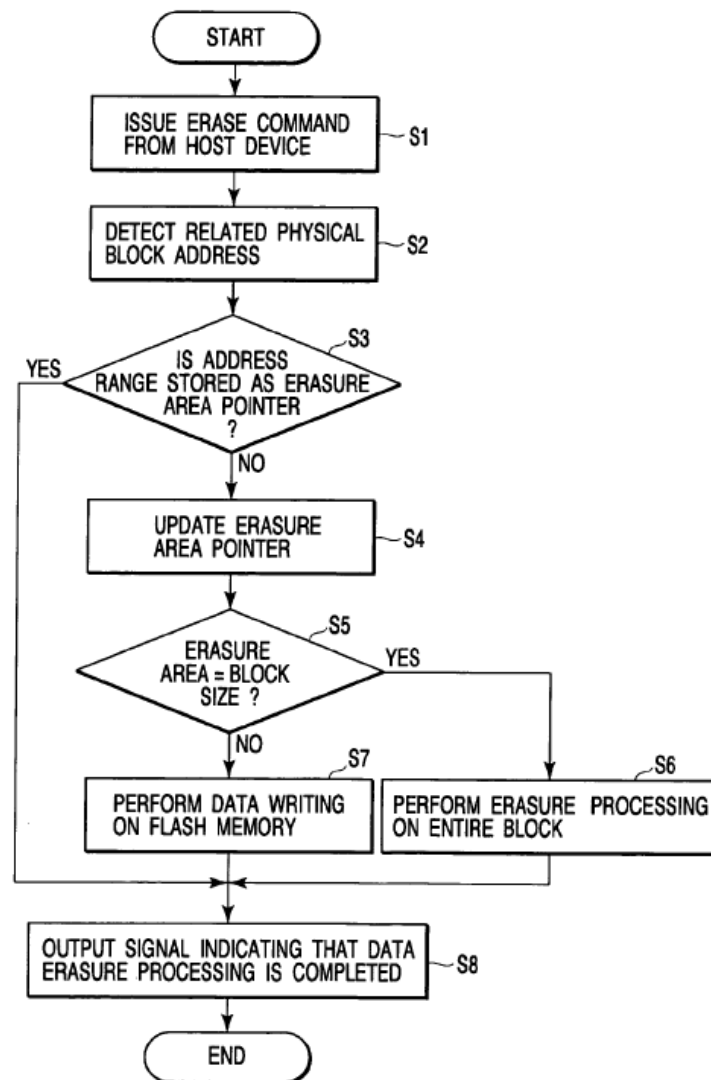


FIG. 8

As indicated above in step S1, host device 2 issues an erase command to erase particular pages of data stored in a physical block. *Id.* at 7:11–19. In step S2, flash memory controlling section 11 refers to logical and physical address table 13a and “detects the physical address of a physical block related to a logical block given an address design[at]ed in the erase command.” *Id.* at 7:30–34.

In step S3, flash memory controlling section 11 “determines whether an address range corresponding to an area in which the data items to be erased in response to the erase command are stored is already stored in the erasure area pointer storage area 13b.” Ex. 1003, 7:38–42. If not, in step S4 it stores erasure area pointers (i.e., a “start pointer” for the first page address and “end pointer” for the last page address) in erasure area pointer storage area 13b. *Id.* at 5:36–46, 7:43–55, Figs. 3–5.

In step S5, flash memory controlling section 11 then “determines whether or not the address range indicated by the erasure area pointer[s] . . . is coincident with the size of a physical block to be subjected to data erasure.” Ex. 1003, 7:56–63. If it is coincident (i.e., the entire physical block is already in a “virtual erased state”), in step S6 the block may be put into an “unused state” by, for example, erasing the associated address information in logical and physical address table 13a. *Id.* 5:54–6:3, 6:34–41, 7:64–8:2, Figs. 4, 6 (physical block B). If it is not coincident, in step S7 flash memory controlling section 11 writes the data items written to erasure area pointer storage area 13b to flash memory 14 so that “even if a power supply to the memory card 1 is turned off, the information of the erasure area pointer is maintained, and thus a virtual erased state is also maintained.” *Id.* at 8:3–12, Figs. 3, 5, 6 (physical blocks A and C).

2. *Independent Claim 1*

For independent claim 1, Petitioner maps the claim limitations to Suda as shown in Petitioner’s annotated version of Suda’s Figure 1, reproduced below:

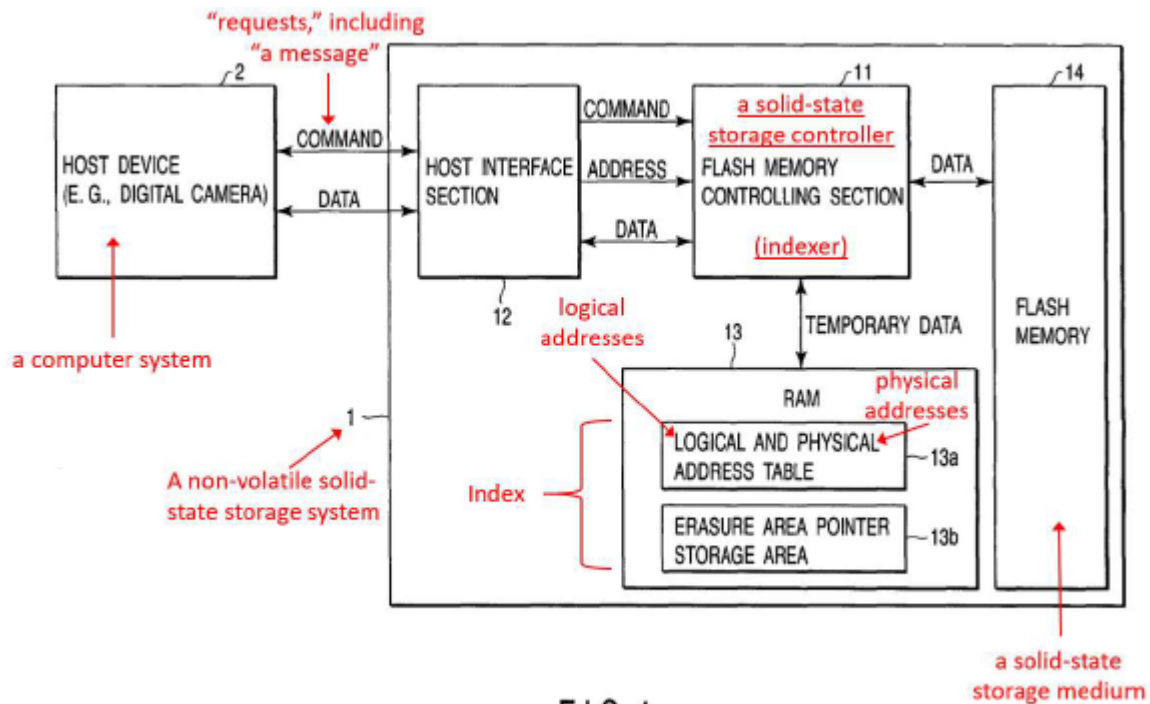


FIG. 1

Pet. 39. In the annotated figure above, red labels indicate the parts of Suda's system that, according to Petitioner, correspond to various elements of claim 1, including the recited "solid-state storage medium" (flash memory 14), the "solid-state storage controller" (flash memory controlling section 11), and the "indexer" (within flash memory controlling section 11). We discuss Petitioner's mapping of claim limitations to the features of Suda in more detail below.

(a) Preamble (1[a]) and Limitation 1[b]

The preamble of claim 1 (1[a]) and limitation 1[b] recite "[a]n apparatus, comprising: a solid-state storage medium." Ex. 1001, 53:22–23. Petitioner contends that Suda discloses an "apparatus" as shown in Figure 1, and that the recited "solid-state storage medium" is nonvolatile flash memory 14. Pet. 39–40 (citing Ex. 1003, 2:63–66, Fig. 1; Ex. 1004 ¶ 185).

UTL does not contest Petitioner’s arguments for the preamble or limitation 1[b], and we find them persuasive because Suda’s Figure 1 clearly discloses an apparatus comprising a solid-state storage medium. *See, e.g.*, Ex. 1003, Fig. 1.

(b) Limitation 1[c]: “a solid-state storage controller
...”

Limitation 1[c] recites “a solid-state storage controller configured to implement storage operations on the solid state storage medium in response to requests from a computer system, including storing data pertaining to logical addresses of a logical address space at respective physical addresses of the solid-state storage medium.” Ex. 1001, 53:24–29.

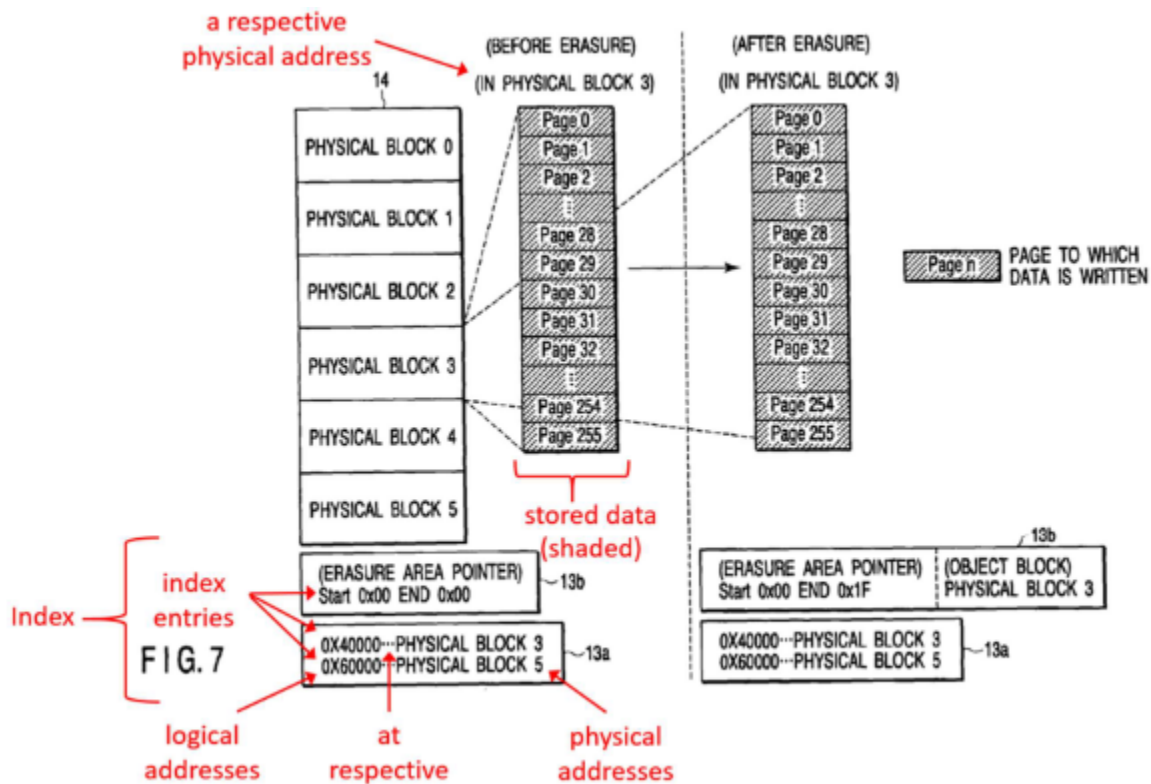
Petitioner argues that Suda discloses this limitation, and UTL does not contest this in its Response. *See* Pet. 40–41; PO Resp. 24–42. In particular, Petitioner argues, and we agree, that the recited solid-state storage controller corresponds to flash memory controlling section 11 as shown in Suda’s Figure 1. Pet. 40 (citing Ex. 1003, 2:63–65, Fig. 1). We also agree with Petitioner that flash memory controlling section 11 implements the recited “storage operations” including reading, writing, and erasing in response to requests from host device 2, and these operations pertain to logical addresses corresponding to physical addresses in flash memory 14. Pet. 40–41 (citing Ex. 1001, 2:41–43, 2:66–3:15, 3:24–27, 3:36–40, 3:43–55, 8:66–9:13, Figs. 1, 2, 7, 11 (step B6); Ex. 1004 ¶¶ 186–188).

Thus, Petitioner has shown that Suda discloses a solid-state storage controller as recited in limitation 1[c].

- (c) Limitation 1[d]: “an indexer . . . configured to assign logical addresses . . . to physical addresses . . .”

Limitation 1[d] recites “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.” Ex. 1001, 53:30–34.

Petitioner argues that Suda discloses the recited “indexer,” which is “the circuitry, software, and/or firmware in Suda’s ‘flash memory controlling section 11 [that] manages data erasure and a table indicating a relationship between logical blocks and physical blocks of the flash memory 14.’” Pet. 41 (alteration in original) (quoting Ex. 1003, 3:13–15) (citing Ex. 1004 ¶ 189; Ex. 1013, 3). In particular, Petitioner provides an annotated version of Suda’s Figure 7, which we reproduce below:



Pet. 42. As described in Suda, Figure 7 depicts an “example of the procedure for erasing data written to the memory card 1,” corresponding to the flowchart depicted in Figure 8 (reproduced above in Section III.C.1). Ex. 1003, 7:5–10. As shown in the red annotations above, Petitioner identifies the recited logical-to-physical address mappings as the entries in Suda’s logical and physical address table 13a. Pet. 41 (citing Ex. 1003, 3:33–55, 3:64–67, Figs. 1, 2, 7; Ex. 1004 ¶¶ 190–191).

In response, UTL argues that rather than flash memory controlling section 11, *other* components of Suda’s device perform the functions of the recited “indexer.” See PO Resp. 24. For example, UTL contends that “there is no discussion of flash memory controlling section 11” in Suda’s description of Figure 2. *Id.* at 24–25 (citing Pet. 41; Ex. 1003, 3:33–40; Ex. 2010 ¶ 82); see also PO Sur-reply 10 (citing Ex. 1003, 3:40–47).

UTL also contends that Petitioner actually relies on RAM 13 for indexing functions rather than flash memory controlling section 11. PO Resp. 25 (citing Pet. 41; Ex. 1003, 3:41–47; 3:64–67, Fig. 2; Ex. 2010 ¶ 83); *id.* at 26–27 (citing Pet. 41, 51–52; Ex. 1003, Fig. 1). According to UTL, “[i]f Suda assigns any logical addresses to physical addresses, the assignments take place in RAM 13 which includes table 13a (the logical and physical address table) but it does not include the flash memory controlling section 11.” *Id.* at 25–26 (citing Ex. 1003, 3:43–47, Fig. 1; Ex. 2010 ¶ 84); *see also* PO Sur-reply 11 (arguing that “Petitioner concedes that table 13a assigns logical block addresses to physical addresses”).

Finally, UTL argues that Petitioner only alleges that flash memory controlling section 11 “detects” logical and physical addresses, but “[d]etecting’ logical and physical addresses that have already been assigned is not ‘assign[ing] logical addresses of the logical address space to physical addresses’ as required by Claim 1.” PO Resp. 27–28 (second alteration in original) (quoting Ex. 2010 ¶ 86); *see also* PO Sur-reply 10 (arguing that Petitioner has not pointed to any disclosure in Suda that “*assigns* anything”).

In its Reply, Petitioner argues that it made clear in its Petition that the recited “indexer” includes the circuitry, software, and firmware in Suda’s flash memory controlling section 11 that *manages* data erasure and a table indicating a relationship between logical blocks and physical blocks of flash memory 14, and that nothing in the claim prohibits an indexer from storing its associated logical-to-physical address table in RAM. Pet. Reply 17 (citing Pet. 41; Ex. 1003, 3:64–67; Ex. 1004 ¶ 190).

We find Petitioner’s argument persuasive and disagree with UTL’s counterarguments. Suda’s Figure 7 appears to show an assignment of logical

addresses (0X40000–0X60000) to physical addresses (blocks 3–5). *See* Ex. 1003, 7:14–17, Fig. 7. Flash memory controlling section 11 *manages* table 13b, including the “relationship between logical blocks and physical blocks.” *Id.* at 3:13–15. During erasure, it is also flash memory controlling section 11 that “cancel[s] the relation between the logical block addresses and the physical addresses” in table 13a, “in order that a physical block . . . to be erased be set in an unused state.” *Id.* at 5:62–6:3; *see also infra* Section III.C.2(d) (discussing the “remove an assignment” aspect of limitation 1[e]).

Thus, although flash memory controlling section 11 (which Petitioner identifies as the “indexer”) *uses* table 13a in RAM 13 to store data indicating the relationship between logical and physical addresses, it is flash memory controlling section 11 that creates and removes the assignments in this table. Table 13a is not the component that makes the assignments: it is merely the medium on which the assignments are recorded. And as Petitioner correctly argues, there is nothing explicit in claim 1 that would prohibit the indexer from using a medium external to the indexer itself. *See* Pet. Reply 17.

Thus, Petitioner has persuasively shown that Suda discloses limitation 1[d].

- (d) Limitation 1[e]: “wherein the indexer is further configured to remove an assignment between an identified logical address and a physical address . . . in response to a message indicating that the identified logical address is erased”

Limitation 1[e] recites “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, the message indicating that the identified logical address is erased.” Ex. 1001, 53:35–40.

Petitioner argues that the indexer in Suda’s flash memory controlling section 11 is configured to perform these limitations. *See* Pet. 42–45. First, Petitioner identifies the recited “message” as an erase command, which is “sent by a host device, such as a digital camera, which a [person of ordinary skill in the art] would have known to be running an operating system.” Pet. 42–43 (citing Ex. 1003, Figs. 1, 8 (step S1); Ex. 1004 ¶ 192). According to Petitioner, this erase command specifically designates a logical address. Pet. 42, 44 (citing Ex. 1003, 6:66–9:3, Fig. 7).

Petitioner points to two examples in Suda that, according to Petitioner, disclose removing the assignment between an identified logical address and a physical address. In the first example, flash memory controller section 11 receives a message marking the beginning and end of a block with erasure area pointers, thus marking the entire block for erasure and “canceling the relation between the logical block addresses and the physical addresses.” Pet. 43 (quoting Ex. 1003, 5:65–6:3) (citing Ex. 1003, Fig. 4; Ex. 1004 ¶ 193). In the second example, flash memory controller section 11 receives a command to erase data spanning a number of blocks including block B, thus removing the relationship between block B and a logical block address. *See* Pet. 43–44 (citing Ex. 1003, 6:15–21, 6:33–41, Fig. 6; Ex. 1004 ¶ 193); *see also* Ex. 1003, 6:36–40 (“[F]lash memory controlling section 11 . . . erases address information of the physical block B and a logical block address related to the address information of the physical block B from the logical and physical address table 13a.”).

Finally, Petitioner contends that a person of ordinary skill in the art “would have understood Suda’s erase command to indicate that ‘the identified logical address is erased’ at the host device” because “the host device is a digital camera that lets users delete, for example, unwanted photos.” Pet. 44 (citing Ex. 1003, 2:61–62, Fig. 1). Citing testimony of Dr. Baker, Petitioner argues that a person of ordinary skill in the art would have understood that a digital camera keeps logical identifiers in cache memory corresponding to stored photos, and when a photo is deleted, the camera erases the cached entry for the photo file; thus, the erase command “indicates that ‘the identified logical address is erased’ in the memory of the digital camera.” Pet. 44–45 (citing Ex. 1003, 8:66–9:3; Ex. 1004 ¶¶ 194–195).

In response, UTL argues that by the time flash memory controlling section 11 receives a command, it is not the same erase command that host device 2’s operating system originally sent. *See* PO Resp. 29 (citing Ex. 2010 ¶ 88). According to UTL, the initial message from device 2 passes through host interface section 12, which “extracts address information.” *Id.* at 30 (quoting Ex. 2010 ¶ 89). Because the “command” sent by host interface section 12 to flash memory controlling section 11 lacks address information, UTL argues that the resulting message it is not the same as the original message when it arrives at flash memory controlling section 11, and thus is not a “message received from a host operating system” as recited in claim 1. *Id.* at 30–32; PO Sur-reply 12–13.

Relying on testimony by Dr. Madisetti, UTL also contends that an ordinarily skilled artisan would have expected, for security reasons, that host interface section 12 would have used different commands for incoming and

outgoing messages, which would have been part of the purpose of having a host interface section. *See* PO Resp. 31–32 (citing Ex. 2010 ¶ 90); PO Sur-reply 13.

In its Reply, Petitioner counters that “as is clear from Suda’s disclosures, Suda discloses only a single ‘command.’” Pet. Reply 18 (citing Ex. 1003, Figs. 1, 8). Petitioner also argues that “nothing in the claim prohibits extracting address information from a command onto a parallel bus,” and “there is no recital in the claims that bars the claimed ‘apparatus’ from processing the received indication into command and address signals.” *Id.* (citing Ex. 1001, claim 1).

We find Petitioner’s arguments persuasive. In particular, we disagree that host interface section 12’s extraction and separate delivery of address information from the erase command converts the original command into a different message. Ultimately, flash memory controlling section 11 receives both the “command” and the extracted address information, so if there is any change, it is to the *form* of the message from host device 2’s operating system, not its content or what the message indicates to the indexer.

Further, Suda explicitly refers to the erase command, received at flash memory controlling section 11, as being “from the host device 2.” Ex. 1003, 7:30; *see also id.* at Fig. 8, step S1 (“ISSUE ERASE COMMAND FROM HOST DEVICE”). And Suda discloses that “the processing [performed by flash memory controlling section 11] is performed in response to *the erase command.*” *Id.* at 7:17–18 (emphasis added). Thus, we agree with Petitioner that a person of ordinary skill in the art would have understood the erase command issued by the operating system of host device 2 to be—despite the

address information being extracted and transmitted separately—the same erase command received by flash memory controlling section 11.

We do not credit Dr. Madisetti’s testimony that an ordinarily skilled artisan “would expect a system such as Suda to use different frontend and backend commands” for security purposes. Ex. 2010 ¶ 90. Dr. Madisetti does not point to any underlying evidence to support this opinion. *See Apple Inc. v. MPH Techs. Oy*, 28 F.4th 254, 262–63 (Fed. Cir. 2022) (the Board may reject expert testimony “based on a lack of factual support” (citing *TQ Delta, LLC v. CISCO Sys., Inc.*, 942 F.3d 1352, 1362 (Fed. Cir. 2019))). And his opinion appears to be contrary to Suda’s reference to a single “erase command” to which flash memory controlling section 11 responds.

Next, UTL argues that a person of ordinary skill in the art would not have understood that Suda’s flash memory controlling section 11 (the proposed “indexer”) removes an assignment between logical and physical addresses, or that any such removal takes place “in response to” the erase message. PO Resp. 33 (citing Ex. 2010 ¶ 93).

In particular, UTL argues that when Suda cancels the relation between a logical block address and a physical block address, “this canceling only occurs in a limited circumstance that is independent of the erase message, namely only when the erasure area pointers span an entire physical block.” PO Resp. 33 (citing Ex. 1003, 5:65–67, claims 3, 11; Ex. 2010 ¶ 94). In particular, UTL points to step S3 in Suda’s Figure 8 in which, “before processing the erase [command], Suda confirms that the physical address corresponding to the logical address provided in the erase command was not previously marked for erasure” in erasure area pointer storage area 13b. *Id.* at 34 (quoting Ex. 2010 ¶ 95). According to UTL, “if Suda’s erase command

removed the mapping between the logical and physical addresses, it would not be possible for a physical address to be previously marked for erasure.” *Id.* (quoting Ex. 2010 ¶ 95); *see also id.* at 35–36 (discussing a similar disclosure in Suda’s Figure 9 regarding the process for reading data).

On the other hand, UTL argues that in the limited circumstance in which a cancelling of the relationship between a logical and a physical address *does* occur (when the erasure area pointers span an entire physical block), this erasure is not “in response [to] a message from the host that the identified logical address is erased, as required by the claims.” PO Resp. 37. Rather, according to UTL, it is “in response to the determination by the flash memory controlling section [11] that the erasure area equals the block size (S5).” *Id.* (citing Ex. 1003, Fig. 8; Ex. 2010 ¶ 97).

Relying on Dr. Madisetti’s testimony, UTL also contends that a person of ordinary skill in the art would have understood that “Suda’s block erasure it not controlled by the host. Rather, the flash memory control system controls block erasure.” PO Resp. 37 (quoting Ex. 2010 ¶ 98) (citing Ex. 1003, 6:33–41). And according to UTL, this block erasure is an “independent decision” that “is not in response to a message received from a host operating system as required by the claims.” *Id.* (citing Ex. 2010 ¶ 98).

We find Petitioner’s arguments persuasive, at least as they relate to the two examples (*see* Figs. 4 and 6) in which one or more entire blocks are erased. We agree with Petitioner that Suda discloses removing an assignment between an identified logical address and a physical address in at least these examples. *See* Ex. 1003, 5:62–6:3, 7:64–8:2. And we disagree with UTL that this erasure is not “in response to” the erase command received from host device 2. Although receiving the erase command is not the sole factor that

could decide whether an entire block is erased, we agree with Petitioner that a person of ordinary skill in the art would have understood that the erasure is “in response to”⁷ the erase command.

UTL argued during the hearing that Suda does not set erasure area pointers “directly” in response to an erase command. *See* Tr. 65:1–4, 71:19–72:15. But claim 1 only states that the removal of an assignment between logical and physical addresses is “in response to” the message from the host operating system, and does not explicitly preclude any intervening steps between when the indexer receives the message (step S1 in Suda’s Figure 8) and when it removes the assignment between the logical and physical addresses (step S6). *See* Ex. 1001, 53:35–40. And we find nothing in the ’727 patent disclosure to suggest that the phrase *in response to* requires a direct causation without any intervening conditions or decision-making. Thus, we disagree with UTL’s contention that step S5 in Suda’s Figure 8 undermines the idea that block erasure in step S6 occurs “in response to” the erase command.

Suda’s usage of the phrase *in response to* is further evidence that we are correctly applying the ordinary and customary meaning of this term. Suda states that the processing shown in Figures 7 and 8 is “performed *in response to* the erase command.” Ex. 1003, 7:17–18 (emphasis added). Suda’s use of the phrase *in response to* in this context mirrors the use of the same phrase in limitation 1[e], and suggests that the ordinary and customary

⁷ We invited the parties to propose constructions for the term *in response to*, but neither party argued that a construction was necessary, and the parties agree that the term has its plain and ordinary meaning. *See* Dec. 32; Pet. Reply 18 (citing Ex. 1038, 32:1–33:6); PO Sur-reply 16 n.3.

meaning of that term does not preclude further decision-making prior to responding to a command from an operating system.

We have considered Dr. Madiseti's testimony that "Suda's block erasure is not controlled by the host. Rather, the flash memory control system controls block erasure." Ex. 2010 ¶ 98 (citing Ex. 1003, 6:33–41). However, claim 1 as applied to Suda does not require that the host controls block erasure—only that the assignment removal is "in response to a message received from a host operating system." Ex. 1001, 53:37–39. The evidence of record shows that in Suda, the erasure of an entire block, and thus the removal of an assignment between logical and physical addresses, occurs in response to receiving the erase message from the operating system of host device 2.

Thus, we determine that Suda discloses limitation 1[e]. And for all reasons discussed in the sections above, we determine that Petitioner has shown by a preponderance of the evidence that claim 1 is unpatentable as obvious over Suda.

3. Dependent Claims 2 and 3

Claim 2 depends from claim 1 and further recites "wherein the indexer assigns logical addresses to physical addresses by use of index entries, and wherein the indexer removes an index entry corresponding to the identified logical address in response to the message." Ex. 1001, 53:41–45. Petitioner argues that in Suda, flash memory controlling section 11 (the proposed "indexer") uses index entries stored in table 13a, which are removed in response to the message. Pet. 45–47 (citing Ex. 1003, 3:13–15, 3:33–40, 3:48–55, 3:64–67, 5:65–6:3, 6:15–25, 6:33–41, 7:32–35, 7:45–51, 8:66–9:3, Figs. 4, 6, 7, 10; Ex. 1004 ¶¶ 197–201).

Claim 3 depends from claim 1 and further recites “wherein the indexer assigns logical addresses to physical addresses by use of index metadata maintained in a memory of the storage controller.” Ex. 1001, 53:46–48. Petitioner identifies RAM 13 as the recited “memory” and the logical and physical addresses in table 13a as the recited “index metadata.” Pet. 47 (citing Ex. 1003, 3:1–2, 3:13–15, 3:41–43, Figs. 1, 7; Ex. 1004 ¶¶ 203–204; Ex. 1013, 3).

For both claims 2 and 3, UTL relies on its arguments for claim 1 that Suda does not disclose an “indexer.” See PO Resp. 38–39. According to UTL, “[b]ecause Suda does not disclose an *indexer* as claimed, it similarly cannot use *index entries*” and “it similarly cannot use index metadata maintained in a memory of the storage controller.” *Id.* (quoting Ex. 2010 ¶ 102).

For the reasons above, we find Petitioner’s argument persuasive that Suda discloses the recited “indexer.” See *supra* Section III.C.2(c). We also find Petitioner’s argument persuasive, and credit Dr. Baker’s supporting testimony, that Suda’s assignments of logical to physical addresses stored in table 13a are “index entries” as recited in claim 2 and are “index metadata maintained in a memory of the storage controller” as recited in claim 3. See Ex. 1003, Fig. 7 (13a); Ex. 1004 ¶¶ 197–204. Thus, Petitioner has shown, by a preponderance of the evidence, that claims 2 and 3 are unpatentable as obvious over Suda.

4. *Dependent Claim 5*

Claim 5 depends from claim 1 and further recites “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.” Ex. 1001, 53:51–55.

Petitioner argues that Suda discloses such a garbage collector in the form of hardware and software for recovering space in Suda’s flash memory. Pet. 48–49 (citing Ex. 1003, 1:54–55, 3:13–15, 5:19–23, 5:40–48, 5:54–6:3, 6:9–41, 6:29–45, 6:60–64, 7:29–34, 7:38–41, 8:66–9:3, Figs. 4–8, 10; Ex. 1004 ¶¶ 209–211).

UTL does not contest Petitioner’s arguments regarding claim 5. *See* PO Resp. 24–42. We find those arguments persuasive, and credit Dr. Baker’s supporting testimony, because Suda discloses a system for recovering unused memory, which is a “garbage collector” as that term would have been understood in the art. *See, e.g.*, Ex. 1003, Figs. 4–7, 10; Ex. 1004 ¶¶ 209–211. Thus, we determine that Petitioner has shown, by a preponderance of the evidence, that claim 5 is unpatentable as obvious over Suda.

5. *Dependent Claim 6*

Claim 6 depends from claim 1 and further recites

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 a universal serial bus interface, an Institute of Electrical and Electronics Engineers 1394 bus interface, an external Serial Advanced Technology Attachment bus interface, a Peripheral Component Interconnect (PCI) bus interface, a PCI Express bus interface, an InfiniBand interface, an Integrated Drive Electronics (IDE) bus interface, an AT Attachment (ATA) interface, a Parallel ATA (PATA) interface, a Serial ATA (SATA) bus interface, an external SATA bus interface, a Small Computer System Interface (SCSI) bus

interface, an internet SCSI interface, and a Fibre Channel interface.

Ex. 1001, 53:56–54:3.

Petitioner contends that Suda discloses a bus interface, although it “does not expressly disclose the types of bus interfaces that may be supported by the host interface section 12.” Pet. 49 (citing Ex. 1003, 3:4–6, Fig. 1; Ex. 1004 ¶ 212). But according to Petitioner, the bus interfaces recited in claim 6 “merely include industry standards that were well-known to a [person of ordinary skill in the art] at the time [of the claimed invention].” *Id.* at 49–50 (citing Ex. 1007, 1, 19 (the SATA 2.5 Standard); Ex. 1015, 92 (a provisional application in the priority chain for the ’727 patent, stating that block device protocols such as SCSI and ATA were “ubiquitous”); Ex. 1004 ¶¶ 213–214).

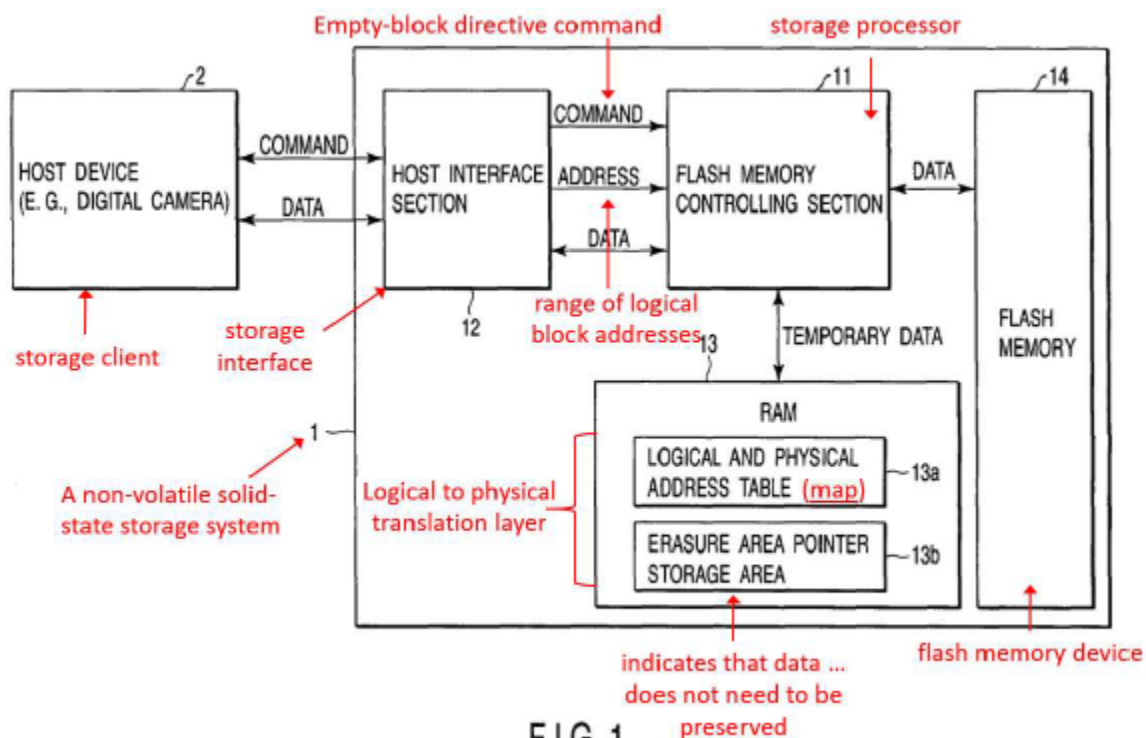
UTL does not contest Petitioner’s arguments regarding claim 6. *See* PO Resp. 24–42. We also note that according to Dr. Baker’s articulation of the level of ordinary skill in the art, which UTL has agreed to for this proceeding, a person of ordinary skill in the art would have been familiar with the ATA standard as part of their background knowledge. *See* Pet. 5; Ex. 1004 ¶ 61; PO Resp. 16; Ex. 2010 ¶¶ 47–52.

We find Petitioner’s arguments persuasive, and credit Dr. Baker’s supporting testimony. Suda discloses a bus interface (host interface section 12), and although Suda does not appear to specify the details of that interface, there were a limited number of well-known bus interfaces at the time of the claimed invention, and a person of ordinary skill in the art would have had reason and ability to select any of them. *See* Ex. 1004 ¶¶ 213–214; Ex. 1015, 92. Thus, we determine that Petitioner has shown, by a

preponderance of the evidence, that claim 6 is unpatentable as obvious over Suda.

6. *Independent Claim 12*

For independent claim 12, Petitioner maps the claim limitations to Suda as shown in an annotated version of Suda's Figure 1, which we reproduce below:



Pet. 51. In the annotated figure above, red labels indicate the parts of Suda's system that allegedly correspond to various elements of claim 12, including the recited "non-volatile solid-state storage system" (memory device 1), the "storage interface" (host interface section 12), the "storage processor" (flash memory controlling section 11), the "flash memory device" (flash memory 14) and the "logical-to-physical translation layer" (logical and physical address table 13a and erasure area pointer storage area 13b). We discuss

Petitioner’s mapping of claim limitations to the features of Suda in more detail below.

(a) Preamble (12[a]) and Limitations 12[b]–[d]

The preamble of claim 12 (12[a]) and limitations 12[b]–[d] recite “[a] non-volatile solid-state storage system, comprising: a storage interface configured to communicate with a storage client; a storage processor coupled to the storage interface; [and] a flash memory device coupled to the storage processor.” Ex. 1001, 54:42–47.

Petitioner does not affirmatively allege that the preamble is limiting, but argues that Suda teaches the preamble and the above limitations. *See* Pet. 50–51. UTL does not contest this. *See* PO Resp. 39–42. We find Petitioner’s arguments persuasive as to the preamble and these limitations, and we credit Dr. Baker’s supporting testimony. In particular, Petitioner argues, and we agree, that Suda discloses a non-volatile solid-state storage system as shown in Figure 1, that the recited “storage interface” is host interface section 12, that the “storage client” is host device 2, that the “storage processor” is flash memory controlling section 11, and that the “flash memory device” is flash memory 14. *See* Pet. 50–52 (citing Ex. 1003, 2:57–3:6, Fig. 1; Ex. 1004 ¶¶ 216–219).

(b) Limitation 12[e]: “a logical-to-physical translation layer . . .”

Limitation 12[e] recites “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.” Ex. 1001, 54:48–53.

Petitioner argues that Suda teaches this limitation. *See* Pet. 52–53. UTL does not contest these arguments. *See* PO Resp. 39–42. We find Petitioner’s arguments persuasive as to limitation 12[e], and we credit Dr. Baker’s supporting testimony. In particular, Petitioner argues, and we agree, that in Suda, the recited “logical-to-physical translation layer . . . includes the logical and physical address table [13a] and the erasure area pointer storage area [13b].” Pet. 52–53 (citing Ex. 1003, 3:13–15, 3:41–55, Figs. 1, 7; Ex. 1004 ¶¶ 221–223).

(c) Limitation 12[f]: “receive . . . an empty-block directive command and a range of logical block addresses”

Limitation 12[f] recites “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.” Ex. 1001, 54:52–56.

Petitioner argues that Suda teaches this limitation. *See* Pet. 53–55. UTL does not contest this. *See* PO Resp. 39–42. We find Petitioner’s arguments persuasive as to limitation 12[f], and we credit Dr. Baker’s supporting testimony. In particular, Petitioner argues, and we agree, that the recited “empty-block directive command”⁸ corresponds to Suda’s “erase command,” which, in some of Suda’s examples (e.g., Figure 6), designates a

⁸ As we discuss above, the district court construed *an empty-block directive command* to mean “[a] command that indicates that certain blocks contain data that does not need to be preserved.” Ex. 2006, 2; *see supra* Section III.B. Both parties have adopted this construction for this proceeding. *See* Pet. 5, 54–55; PO Resp. 16–17.

range of logical block addresses. Pet. 54–55 (citing Ex. 1003, 3:4–12, 6:15–26, 7:30–34, 8:66–9:3, Fig. 1; Ex. 1004 ¶¶ 224–228).

(d) Limitation 12[g]: “update the logical-to-physical translation layer . . .”

Limitation 12[g] recites “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.” Ex. 1001, 54:57–60.

Petitioner argues that Suda teaches this limitation. *See* Pet. 55. UTL does not contest this. *See* PO Resp. 39–42. We find Petitioner’s arguments persuasive as to limitation 12[g], and we credit Dr. Baker’s supporting testimony. In particular, Petitioner argues, and we agree, that “Suda teaches to use erasure area pointers to mark data at the designated addresses as in a ‘virtual erased state,’ and is ‘to be deleted,’ meaning the data will not be preserved.” Pet. 55 (citing Pet. 42–45; Ex. 1003, 5:40–48, 5:57–61, 6:9–14, 6:29–45, 6:60–64, 7:38–41, Fig. 8 (updating at step S4); Ex. 1004 ¶ 230).

(e) Limitation 12[h]: “store persistent data on the flash memory device . . .”

Limitation 12[h] recites “wherein the storage processor is configured to . . . 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.” Ex. 1001, 54:61–64.

Petitioner illustrates its contentions regarding limitation 12[h] in an annotated version of Suda’s Figure 1, which we reproduce below:

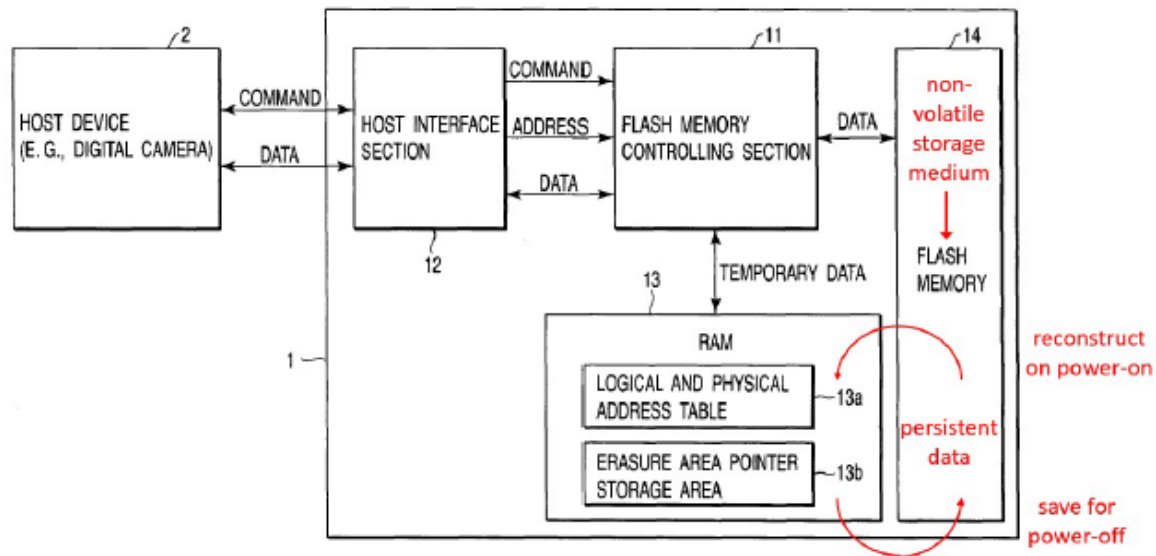


FIG. 1

Pet. 56. As shown in red annotations above, Petitioner identifies the recited “persistent data” as being within flash memory 14, and argues that any data stored within flash memory 14 is “persistent data” because it is nonvolatile. Pet. 55 (citing Ex. 1003, 2:65–66). Petitioner adds two arrows: (1) an arrow labeled “save for power-off” from erasure area pointer storage area 13b to the persistent data in flash memory 14, and (2) an arrow labeled “reconstruct on power-on” from the persistent data in flash memory 14 to logical and physical address table 13a. This reflects Petitioner’s argument that “Suda stores a copy of its erasure area pointers in flash memory to avoid losing the erasure area pointers when the power supply is turned off.” Pet. 55 (citing Ex. 1003, 8:3–16).

According to Petitioner, erasure area pointers in erasure area pointer storage area 13b “indicate data that is no longer in use, and, ‘[f]rom the user’s perspective, this data has been deleted from a document.’” Pet. 57 (quoting Ex. 1013, 7 (UTL’s infringement contentions)) (citing Ex. 1004 ¶¶ 234–235). Specifically, Petitioner contends that Suda discloses that the

host device is a digital camera, and that a person of ordinary skill in the art “would have known that, when a user selects a photo to delete, the digital camera will look up the logical address corresponding to the digital photo selected for deletion and issue an erase command to the memory device . . . designating the logical address.” *Id.* (citing Ex. 1003, 2:61–62, 8:66–9:3, Fig. 1; Ex. 1004 ¶¶ 235–236). Afterwards, the erasure area pointers “mark this data as in a ‘virtual erased state,’ which indicates that this data is no longer in use [and] should appear deleted from the user’s perspective.” *Id.* (citing Ex. 1003, 5:14–27, 8:21–41, Fig. 9; Ex. 1004 ¶¶ 235–237).

In response, UTL contends that the recited persistent data “must be stored ‘in response to every empty-block directive command or else the claim makes no sense.’” PO Resp. 39 (quoting Ex. 2010 ¶ 103). This is because, according to UTL, “[i]f no empty-block directive command is received, it would make no sense to store persistent data as there would be no received logical block addresses.” *Id.* at 40 (quoting Ex. 2010 ¶ 103).

UTL also contends that “the erasure area pointer data is stored as persistent data only after several steps after the erase command is issued including the determination of whether the address range is coincident with the physical block to be subjected to data erasure which changes what logical block addresses are referenced.” *Id.* (quoting Ex. 2010 ¶ 104). “Thus, the alleged persistent data is not stored in response to or as a result of the empty-block directive command and related received logical block addresses, but after the flash memory controlling section 11 rewrites the data and receives a read command.” *Id.* at 41 (quoting Ex. 2010 ¶ 105); *see also* PO Sur-reply 18–19 (citing Ex. 1003, 8:66–9:14, 9:18–20) (arguing that any

logical block address that accompanies an erase command is changed during the rewriting step).

UTL also argues that “Suda does not disclose that the same ‘range of logical block addresses’ that accompany the ‘empty-block directive command’ is the same as the ‘received logical block addresses’ later in the claim.” PO Resp. 42 (quoting Ex. 2010 ¶ 106).

In its Reply, Petitioner disagrees that claim 12 requires writing persistent data in response to every empty-block directive command. Pet. Reply 21. According to Petitioner, claim 12 “contains no such language and UTL’s argument has no basis in the claim,” and “the claim contains no language that excludes intervening steps, and instead only requires that the storage occur in response to the command.” *Id.* Thus, Petitioner argues that “the persistent data stored in step S7 [when the address range to be erased does not constitute an entire block] corresponds to the logical address received at S1 and determined at S2.” *Id.* at 21–22.

Petitioner also disagrees with UTL that Suda fails to disclose that the persistent data corresponds to the same range of logical block addresses that accompany the empty-block directive command. Pet. Reply 21. For establishing this relationship, Petitioner points to Suda’s Figure 8, step S2, “where the physical block address in Suda’s erase command is detected.” *Id.* (citing Ex. 1003, 7:7–8:12).

In its Sur-reply, UTL argues that Suda’s description of step S2 “does not even reference a logical block address let alone a ‘range of logical block addresses.’” PO Sur-reply 19. UTL also argues that the data writing in step S7 “occurs only after a separate write command is issued and only if the logical block address range is not coincident with the size of the physical

block to be subjected to data erasure.” *Id.* at 20 (citing Ex. 1003, 8:3–11; Ex. 2010 ¶¶ 103–106).

Having considered the arguments and evidence on both sides, we find Petitioner’s arguments persuasive that Suda discloses limitation 1[h]. In particular, we agree that Suda’s storage processor (flash memory controlling section 11) is configured to store persistent data in flash memory 14 (e.g., step S7 of Figure 8) indicating that data corresponding to a range of logical block addresses (received with the erase command in step S1) is deleted at the storage client. *See, e.g.*, Ex. 1003, 6:18–26, 7:30–34, 8:66–9:3, Fig. 6.

We disagree with UTL’s contention that claim 12 requires the storage processor to store persistent data in response to *every* empty-block directive command. Claim 12 only requires that the storage processor is “configured to” store persistent data indicating that the data corresponding to the received logical block addresses is deleted at the storage client—not that it must store persistent data in response to every empty-block directive command. Ex. 1001, 54:53, 54:61–64. Also, we find nothing in claim 12 or the ’727 patent as a whole requiring that the step of storing persistent data must be directly in response to the empty-block directive command, with no intervening decision steps.

We also disagree with UTL that the range of logical block addresses used in writing to flash memory in step S7 is a different range of logical block addresses received with the erase command in step S1. Although UTL points to an example in Suda in which flash memory controlling section 11 performs “rewriting processing” to revise the data in tables 13a and 13b, Suda’s disclosure does not suggest that this rewriting entails changing any logical block addresses. *See* Ex. 1003, 8:66–9:14, 9:18–22. To the contrary,

after the erasure operation, tables 13a and 13b indicate which particular data in the physical blocks, corresponding to the received logical block addresses, have been erased. *See id.*; *id.* at 9:23–34, Fig. 10 (13a, 13b).

For the above reasons, we determine that Petitioner has shown, by a preponderance of the evidence, that claim 12 is unpatentable as obvious over Suda.

7. *Dependent Claim 13*

Claim 13 depends from claim 12 and further recites “wherein the logical-to-physical translation layer is stored in the flash memory device.” Ex. 1001, 53:41–45.

Petitioner argues that “Suda teaches to store its erasure area pointers in the flash memory device in order to prevent data loss when powered off,” and “[i]t would have been obvious . . . that the rest of Suda’s logical-to-physical translation layer, including the logical and physical address table, also be similarly stored in persistent memory for the same reason.” Pet. 57–58 (citing Ex. 1004 ¶¶ 239–241). In other words, Petitioner argues that a person of ordinary skill in the art would have had reason, in light of that person’s background knowledge, to apply Suda’s teachings about persistently storing erasure area pointer storage area 13b to logical and physical address table 13a.

UTL does not contest Petitioner’s arguments regarding claim 13 based on Suda alone, but UTL challenges Petitioner’s arguments based on the combination of Suda and Bennett. *See* PO Resp. 24–44. Because we determine below that Petitioner has shown that claim 13 would have been obvious over the combination of Suda and Bennett, we need not decide

whether claim 13 would have been obvious over Suda alone. *See infra* Section III.D.3.

8. *Dependent Claim 14*

Claim 14 depends from claim 12 and further recites “a volatile memory device coupled to the storage processor, wherein the logical-to-physical translation layer is stored in the volatile memory device.” Ex. 1001, 55:1–4. Petitioner argues that in Suda, this volatile memory device corresponds to RAM 13. Pet. 58 (citing Ex. 1003, Fig. 1; Ex. 1004 ¶¶ 242–244).

UTL does not contest Petitioner’s arguments regarding claim 14. *See* PO Resp. 24–42. We find Petitioner’s arguments persuasive, and credit Dr. Baker’s supporting testimony, because the evidence shows that Suda’s RAM 13, which includes logical-to-physical translation layer 13a, is a volatile memory device. Thus, we determine that Petitioner has shown, by a preponderance of the evidence, that claim 14 is unpatentable as obvious over Suda.

9. *Dependent Claims 15 and 16*

Claim 15 depends from claim 12 and further recites “wherein the storage processor 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.” Ex. 1001, 55:5–9.

Claim 16 depends from claim 15 and further recites “wherein data bits of the predetermined data string have a uniform logic level.” Ex. 1001, 55:10–11.

Petitioner argues that Suda discloses the limitation of claim 15 because if a read request specifies that an address is included in erasure area pointer storage area 13b, flash memory controlling section 11 outputs “initial-value data,” which a person of ordinary skill in the art would have understood to be “a string of empty values.” Pet. 59 (citing Ex. 1001, 1:37–39, 1:56–60 (describing background of the ’727 patent); Ex. 1003, 3:58–59, 4:4–6, 8:21–41, Fig. 9; Ex. 1004 ¶¶ 245–246).

In particular, as to claim 16, Petitioner argues that the string of empty values would be “typically all 1’s.” Pet. 59–60 (citing Ex. 1001, 1:56–60 (describing background of the ’727 patent); Ex. 1003, 3:58–59, 4:4–6; Ex. 1004 ¶¶ 247–248).

UTL does not contest Petitioner’s arguments regarding claims 15 or 16. *See* PO Resp. 24–42. We find Petitioner’s arguments persuasive because Suda clearly discloses returning initial-value data (*see* Ex. 1003, 3:58–59, 4:4–6), and we credit Dr. Baker’s testimony that a person of ordinary skill in the art would have understood that initial-value data comprises a string of uniform values, typically 1’s. *See* Ex. 1004 ¶¶ 73–74, 247 (citing Ex. 1011, 533). Thus, we determine that Petitioner has shown, by a preponderance of the evidence, that claims 15 and 16 are unpatentable as obvious over Suda.

D. OBVIOUSNESS GROUNDS BASED ON SUDA AND BENNETT (CLAIMS 4 AND 13)

Petitioner’s third ground asserts that claims 4 and 13 are unpatentable under § 103(a) as obvious over the combination of Suda and Bennett. Pet. 60–62. Before addressing Petitioner’s arguments, we present a brief overview of Bennett.

1. Bennett

Bennett discloses various “improvements for managing erase processes in non-volatile memory.” Ex. 1002, code (57), 1:17–19. In its background section, Bennett states that in “more advanced memory systems” of that time period, it was common for a non-volatile memory not to immediately erase data when it becomes obsolete, but for the obsolete memory “to be ‘logically erased’ by being marked for erase, with the actual, physical erase taking place at a later time.” Ex. 1002, 3:26–32. The actual erasure could take place when “the memory has time, or is [in] need of additional erased sectors, . . . for example in a background process.” *Id.* at 3:38–41.

Bennett’s disclosure seeks to reduce the time it takes to erase data in a non-volatile memory by using the system’s “standard, logical erase method” for specified sectors of the memory that do not form a complete block. Ex. 1002, 6:13–22. Bennett’s Figure 2, which we reproduce below, is a block diagram illustrating the organization of the logical and physical block structures of such a memory system and its relation to a host device. Ex. 1002, 5:15–18, 7:13–18.

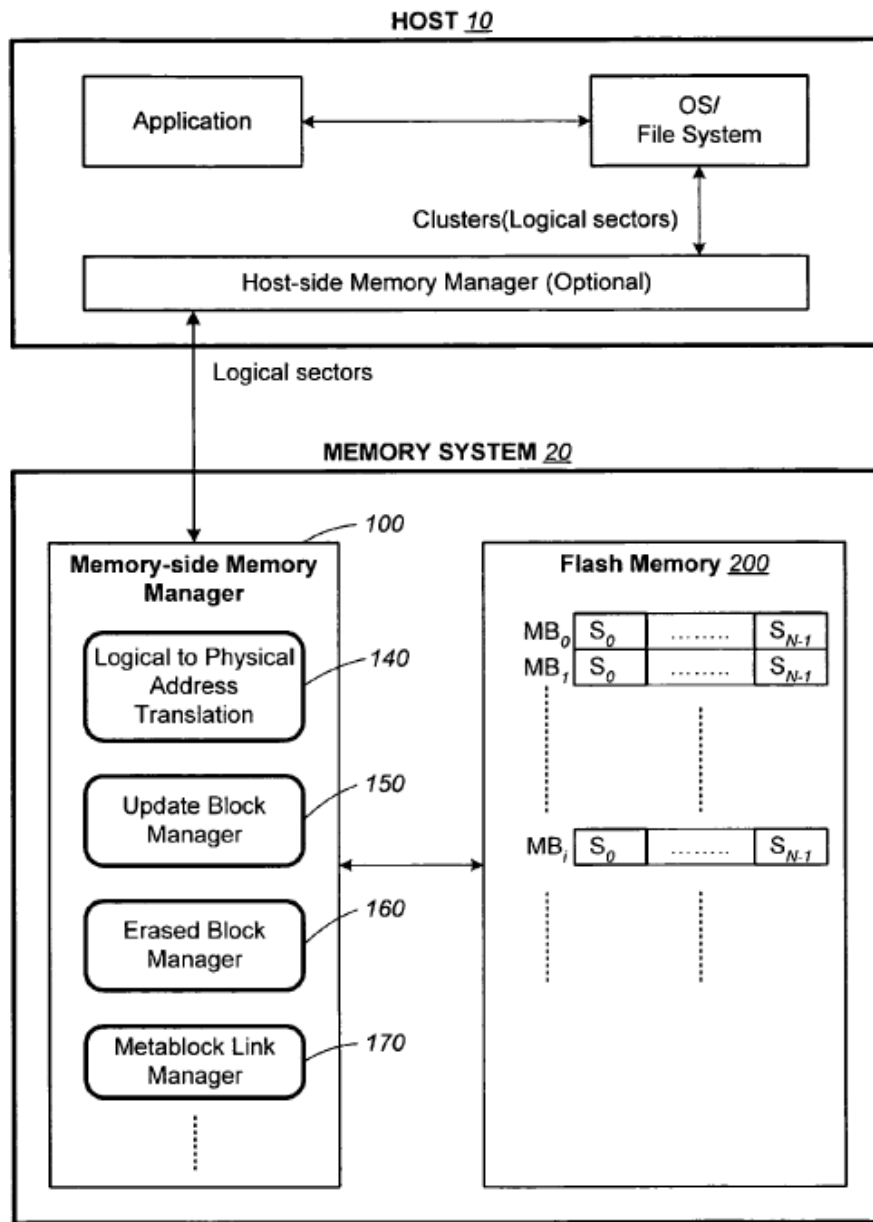


FIG. 2

As shown above, Figure 2 depicts host 10 in communication with memory system 20 comprising memory-side memory manager (controller) 100 and flash memory 200, which is organized into metablocks MB_0 – MB_i . *Id.* at 7:15–19. Each metablock is a group of physical sectors S_0 – S_{N-1} that are erasable together. *Id.* at 7:19–20.

“[H]ost 10 accesses the memory 200 when running an application under a file system or operating system.” Ex. 1002, 7:21–22. “In most cases during read or write operations, the host 10 essentially issues a command to the memory system 20 to read or write a segment containing a string of logical sectors of data with contiguous addresses.” *Id.* at 7:29–32. Flash memory 200 stores a group address table (GAT) that “keeps track of the mapping between logical groups of sectors and their corresponding metablocks.” *Id.* at 10:16–21, Fig. 6.

2. Claim 4

Claim 4 depends from claim 1, and further recites “wherein the indexer comprises firmware of the solid-state storage controller.” Ex. 1001, 53:49–50. Petitioner concedes that “Suda does not explicitly show the internal components of its flash memory controlling section or state that the indexer in the flash memory controlling section includes ‘firmware.’” Pet. 60 (citing Ex. 1003, Fig. 1). So Petitioner cites Bennett for this limitation, arguing that Bennett describes “typical internal components, including ROM, in a controller,” and teaches that “[f]irmware stored in nonvolatile ROM . . . and/or the optional nonvolatile memory . . . provides codes for the processor . . . to implement the functions of the controller 100.” Pet. 60 (citing Ex. 1002, 7:4–7, 24:19–20). According to Petitioner, “[i]t would have been obvious to a [person of ordinary skill in the art] that the indexer in Suda’s ‘flash memory controlling section [11]’ would have had similar internal components to those described in Bennett, including firmware.” *Id.* at 60–61 (citing Ex. 1002, 7:4–7; Ex. 1004 ¶¶ 207–208).

In its Response, UTL first argues that a person of ordinary skill in the art would not have been motivated to combine Suda with Bennett because

“Suda and Bennett describe different concepts.” PO Resp. 43–44. According to UTL, Suda is directed to a “memory management device that uses a solid-state memory in relation to a digital camera involving erase functions which includes a setting unit that predetermines a range of addresses of data to be erased and a controlling unit that outputs data to be read in response to a read command.” *Id.* (citing Ex. 1003, code (57)). By contrast, UTL argues that “Bennett is directed to improving and managing erase processes in nonvolatile memory, but does not disclose any camera system nor any specific device or application,” or any “setting unit or controlling unit as described in Suda that is integral to the invention.” *Id.* at 44 (citing Ex. 1002, code (57); and then quoting Ex. 2010 ¶ 107); *see also* PO Sur-reply 20 (arguing that “Suda and Bennett . . . address different technologies”).

In its Reply, Petitioner disagrees with UTL that Suda and Bennett relate to different concepts or technologies, as “they both relate to techniques for improving flash memory management.” Pet. Reply 22 (citing Ex. 1002, code (54); Ex. 1003, code (57)). Petitioner also disputes the relevance of UTL’s argument that Bennett does not specifically teach a camera embodiment or a specific host device or application, because “Bennett’s figure 2 specifically shows that the erase command comes from an ‘application’ in another ‘host’ device—just like the digital camera in Suda that is also labeled as a ‘host device.’” *Id.* (citing Ex. 1002, Fig. 2; Ex. 1003, Fig. 1).

We find Petitioner’s arguments persuasive. In particular, we credit Dr. Baker’s supporting testimony that an ordinarily skilled artisan would have looked to Bennett for the teaching that the indexer comprises firmware. *See* Ex. 1004 ¶¶ 206–208. Suda does not disclose explicitly what flash

memory controlling section 11 is, but describes what functions it performs in the disclosed memory management device. Specifically, flash memory controlling section 11 is “connected to the host device 2 by the host interface section 12” and “connected to a RAM 13 and the flash memory 14,” “operates based on” the information received from host interface section 12, and “manages data erasure” using tables 13a and 13b stored in RAM 13. Ex. 1003, col. 2, l. 66–col. 3, l. 3, col. 3, ll. 9–15. Bennett discloses a similar memory controller with firmware performing similar functions. Ex. 1004 ¶ 206. A person of ordinary skill in the art would have been well aware of the benefits of using firmware in such a device (e.g., speed and ability to reprogram), which would be “between” those of hardware and software. *Id.* ¶ 207.

We also agree with Petitioner that Suda and Bennett are in the same general field and are directed to similar problems. But even if Suda and Bennett were directed to different concepts, Petitioner persuasively relies on Bennett “as evidence of two basic, common, and obvious techniques: the use of firmware (claim 4) and caching between volatile and persistent storage (claim 13),” and “[n]either technique depends on any alleged distinction between Suda and Bennett or the specific disclosure of a camera.” Pet. Reply 22–23 (citing Ex. 1004 ¶¶ 205–208, 240–241).

Next, UTL argues that “Suda does not disclose firmware at all,” and that Dr. Baker, in his discussion about combining Suda with Bennett, “does not explain how a [person of ordinary skill in the art] would incorporate such firmware into Suda.” PO Resp. 45 (citing Ex. 1004 ¶¶ 206–207; Ex. 2010 ¶ 109). UTL also characterizes Dr. Baker’s testimony as only opining that an ordinarily skilled artisan *could* have used firmware, but not that the artisan

would have been motivated to used firmware. *See id.* at 46–47 (citing Ex. 2010 ¶¶ 111–113); *see also* PO Sur-reply 20 (arguing that “Petitioner fails to even identify what a [person of ordinary skill in the art] would do specifically apart from what they could do”).

In its Reply, Petitioner counters that incorporating firmware into an indexer, as Bennett teaches, would have been within the ordinary skill in the art, and that Dr. Baker explained the benefits of using firmware. Pet. Reply 23–24 (citing Ex. 1004 ¶ 207). Petitioner argues that, to a person of ordinary skill in the art, “programming firmware to operate as an indexer as described by Suda would not have been any more difficult than programming software for Suda’s controller, and this would not have deterred the combination.” Pet. Reply 23. On this point, we credit Dr. Baker’s testimony that when designing an indexer, a person of ordinary skill in the art would have had three well-known choices—hardware, firmware, and software—and that choosing from among these three is a non-inventive matter of weighing the tradeoff between speed and cost. Ex. 1004 ¶ 207. According to Dr. Baker, firmware was understood to reflect a compromise between hardware (the fastest but most expensive) and software (the slowest but cheapest). *See id.*

Thus, based on the evidence of record, we determine that a person of ordinary skill in the art would have had reason to incorporate firmware into Suda’s indexer, and that doing so would have been within the level of ordinary skill in the art. And, for the above reasons, Petitioner has shown, by a preponderance of the evidence that claim 4 is unpatentable as obvious over Suda in light of Bennett.

3. *Claim 13*

Claim 13 depends from claim 12 and further recites “wherein the logical-to-physical translation layer is stored in the flash memory device.” Ex. 1001, 54:65–67.

As we discuss above, Petitioner argues that Suda alone teaches claim 13, but we do not reach that argument. *See supra* Section III.C.7. Alternatively, Petitioner argues in this ground that a person of ordinary skill in the art would have looked to Bennett’s teaching that “mappings between logical groups and physical groups . . . are stored in a set of table[s] and lists distributed among the nonvolatile flash memory 200 and the volatile but more agile RAM.” Pet. 61 (first alteration in original) (quoting Ex. 1002, 10:40–43). According to Petitioner, “[t]his implements a common technique called ‘caching,’ where the logical to physical address table is stored in RAM for faster access and also stored in nonvolatile flash so it is not lost when powered off.” *Id.* (citing Ex. 1004 ¶ 240). Petitioner asserts that a person of ordinary skill in the art “would have found it obvious that Suda’s logical and physical address table could be cached in the same way, such that it is also stored in nonvolatile flash memory in addition to RAM.” *Id.* at 61–62 (citing Ex. 1004 ¶ 240).

UTL’s argument regarding claim 13 goes to the motivation to combine the references, and is the same general argument we discuss in the context of claim 4, above. *See* PO Resp. 42–44; *supra* Section III.D.2. As we explain above, we find Petitioner’s arguments persuasive as to the motivation to combine Suda with Bennett. *See supra* Section III.D.2.

We also find Petitioner’s arguments persuasive that the combination of Suda and Bennett teaches the subject matter of claim 13, and we credit the

supporting testimony of Dr. Baker that a person of ordinary skill in the art would have understood that storing data in flash memory “merely applies a basic principle of data storage devices: to store, and not lose, data,” and would have looked to Bennett for storing a logical-to-physical address table in both RAM and flash memory. Ex. 1004 ¶¶ 240–241.

Thus, we determine that Petitioner has shown, by a preponderance of the evidence, that claim 13 is unpatentable as obvious over Suda in light of Bennett.

E. OBVIOUSNESS GROUNDS BASED ON BENNETT (CLAIMS 1–6 AND 12–16)

In addition to the above grounds, Petitioner asserts that claims 1–6 and 12–16 are unpatentable under § 103(a) as obvious over Bennett alone. Pet. 9, 17–38. As explained above, we conclude that claims 1–6 and 12–16 are unpatentable on the grounds based on Suda or the combination of Suda and Bennett. *See supra* Sections III.C–D.

Because our conclusions above are fully dispositive as to all challenged claims, we need not consider the ground based on Bennett in our decision. *See Boston Scientific Scimed, Inc. v. Cook Group Inc.*, 809 F. App’x 984, 990 (Fed. Cir. 2020) (holding that once a petitioner has prevailed on all its challenged claims, “the Board need not address issues that are not necessary to the resolution of the proceeding”); *Beloit Corp. v. Valmet Oy*, 742 F.2d 1421, 1423 (Fed. Cir. 1984) (once a dispositive issue is decided, there is no need to decide other potentially dispositive issues).

IV. CONCLUSION⁹

For the reasons above, Petitioner has shown by a preponderance of the evidence that claims 1–6 and 12–16 of the '727 patent are unpatentable under § 103(a) as obvious over Suda (claims 1–3, 5, 6, 12, and 14–16), and over the combination of Suda and Bennett (claims 4 and 13). Because this disposition addresses all challenged claims, we do not reach the question of whether any claims are unpatentable on alternative grounds raised in the Petition.

V. ORDER

In consideration of the foregoing, it is
ORDERED that claims 1–6 and 12–16 of the '727 patent are
unpatentable; and

FURTHER ORDERED that parties to this proceeding seeking judicial review of our decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

⁹ Should UTL wish to pursue amendment of claims in a reissue or reexamination proceeding after this decision, we draw UTL's attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If UTL chooses to file a reissue application or a request for reexamination of the challenged patent, we remind UTL of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. § 42.8(a)(3), (b)(2).

In summary:

Claims	35 U.S.C. §	Reference(s)/ Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1–6, 12–16	103(a)	Bennett ¹⁰		
1–3, 5, 6, 12–16	103(a)	Suda	1–3, 5, 6, 12, 14–16	
4, 13	103(a)	Suda, Bennett	4, 13	
Overall Outcome			1–6, 12–16	

¹⁰ As explained above, given our disposition of the grounds based on Suda and the combination of Suda and Bennett, we do not reach Petitioner’s alternative ground asserting that claims 1–6 and 12–16 are unpatentable over Bennett. *See supra* Section III.E. Nor do we reach Petitioner’s argument that claim 13 would have been obvious over Suda alone. *See supra* Section III.C.7.

IPR2021-00345
Patent 9,632,727 B2

For PETITIONER:

Louis L. Campbell
Michael Rueckheim
WINSTON & STRAWN LLP
llcampbell@winston.com
mrueckheim@winston.com

For PATENT OWNER:

Barry Bumgardner
Robert Delafield II
NELSON BUMGARDNER ALBRITTON P.C.
barry@nelbum.com
bobby_delafield@yahoo.com