

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

HEWLETT PACKARD ENTERPRISE CO.,
Petitioner,

v.

INTELLECTUAL VENTURES II LLC,
Patent Owner.

IPR2022-00116
Patent 7,882,320 B2

Before KEN B. BARRETT, NABEEL U. KHAN, and
STEPHEN E. BELISLE, *Administrative Patent Judges*.

KHAN, *Administrative Patent Judge*.

DECISION

Denying Institution of *Inter Partes* Review
35 U.S.C. § 314, 37 C.F.R. § 42.4

I. INTRODUCTION

On October 29, 2021, Hewlett Packard Enterprise Co. filed a Petition requesting *inter partes* review of claims 1–3 and 5–8 (the “challenged claims”) of U.S. Patent No. 7,882,320 B2 (Ex. 1001, the “’320 patent”). Paper 1 (“Pet.”). Intellectual Ventures II LLC (“Patent Owner”) filed a Preliminary Response. Paper 8 (“Prelim. Resp.”). With our authorization, Petitioner filed a Reply to Patent Owner’s Preliminary Response (Paper 10, “Reply”), and Patent Owner filed a Sur-Reply (Paper 11, “Sur-Reply”).

We have authority to institute an *inter partes* review only if the information presented in the Petition shows “there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018). Upon consideration of the papers and evidence of record, we conclude that Petitioner has not shown a reasonable likelihood of prevailing against at least one of the challenged claims. For the reasons explained below, we do not institute an *inter partes* review on the challenged claims of the ’320 patent.

II. BACKGROUND

A. Related Proceedings

The parties originally identified the following pending district court proceeding involving the ’320 patent: *Intellectual Ventures I et al. v. Hewlett Packard Enterprise Co.*, Case No. No. 6:21-CV-00596 (W.D. Tex.)¹ (“parallel litigation”). Pet. 1; Paper 5, 1. Later, Petitioner states, in that parallel litigation, Patent Owner filed an amended complaint which dropped

¹ Patent Owner subsequently updated its Mandatory Notices to explain that the case has been ordered to be transferred to the District of Northern California. Paper 7, 1.

Patent Owner's infringement allegations involving the '320 patent. Paper 6, 1.

Additionally, Patent Owner states that the '320 patent claims the benefit of U.S. Patent No. 7,424,587 B2, U.S. Patent No. 7,461,229 B2, and U.S. Patent No. 7,464,240 B2. Paper 5, 1. Petitioner states that it has filed IPR2022-00115 challenging claims 1, 3, 7, and 8 of U.S. Patent No. 7,464,240 B2. Pet. 1.

B. Real Parties in Interest

Petitioner identifies itself, "Hewlett Packard Enterprise Co.," as the real party-in-interest. Pet. 1. Patent Owner identifies itself, "Intellectual Ventures II LLC," as the real party-in-interest. Paper 5, 1.

C. The '320 Patent

The '320 patent is titled "Multi-processor Flash Memory Storage Device and Management System." Ex. 1001, code (54). The '320 patent discloses a "a multiple processor flash-based storage device" and management of data storage on the device. *Id.* at 4:19–21. Figure 2, reproduced below, "is a block diagram illustrating a multi-processor Flash storage device." *Id.* at 4:1–3.

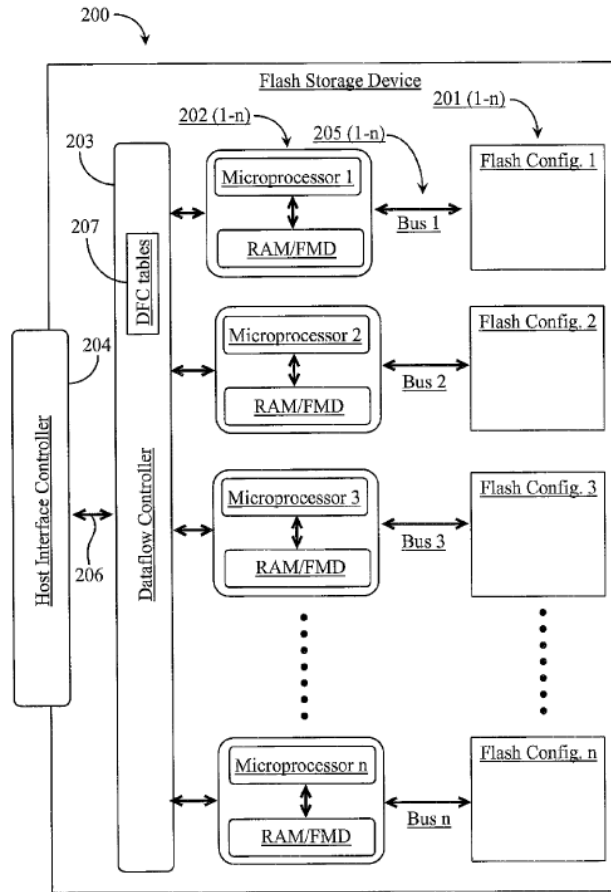


Fig. 2

As shown in Figure 2, multi-processor Flash storage device 200 includes “a distributed processor architecture comprising multiple microprocessor units 202 (1-n)” where “[e]ach microprocessor unit 202 (1-n) includes a microprocessor and an onboard or bused access to a dedicated amount of RAM.” *Id.* at 5:44–48. The ’320 patent explains that the “illustration of separate RAM/[Flash management Data tables (“FMD”)] in each processor unit 202(1-n) is not meant to indicate that there are completely separate and autonomous RAM units, but simply that each microprocessor unit has a dedicated portion of RAM.” *Id.* at 5:54–57. The ’320 patent states that the RAM can be used as a cache and “[o]ne case for using RAM has a cache memory for parking Flash data for eventual write to Flash.” *Id.* at 7:1–3; *see id.* at 10:38. Further shown in Figure 2,

“[m]icroprocessor unit 202(1) is coupled to Flash configuration 1 (Flash device) by dedicated Bus 1. Microprocessor unit 202(2) is coupled to Flash configuration 2 by dedicated Bus 2, and so on.” *Id.* at 6:7–10. Still further shown in Figure 2, “[e]ach microprocessor unit 202 (1–n) has a dedicated bus connection to a unique dataflow controller 203. Dataflow controller 203 manages the data traffic over all of the Flash channels through each of the microprocessor units.” *Id.* at 6:15–18. The ’320 patent notes that “[d]ataflow controller 203 may be a state machine implemented in software or firmware. Also, dataflow controller 203 may be implemented as processor-controlled hardware.” *Id.* at 6:52–54.

The ’320 patent also describes a process for storing data in the multi-processor Flash device, i.e., a write process. *Id.* at 9:62–64. Figure 5, reproduced below, is a process flow chart illustrating steps for the write process. *Id.*

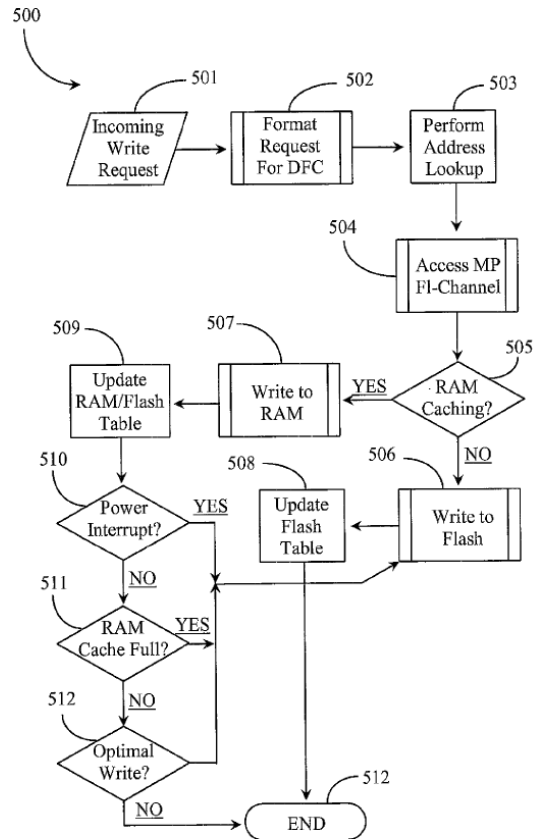


Fig. 5

As shown in Figure 5, first an incoming write request is received by the multi-processor Flash storage device in step 501. *Id.* at 9:64–67. In the next series of steps, the dataflow controller of the multi-processor Flash storage device “determine[s] where the write will occur,” accesses the correct microprocessor unit(s) and Flash memory, and “determines if RAM caching for Flash writes is turned on.” *Id.* at 10:5–10. In the case that RAM caching is turned on, “the system may make a determination at step 511 if there are any full RAM caches on any of the microprocessor units on the Flash storage device.” *Id.* at 10:47–50. That “determination is ongoing as dedicated RAM on one microprocessor unit may enter a full state at any time.” *Id.* at 10:50–53. Further, “[a]s data is overwritten or aged out of validity RAM cache may be purged and periodic Writes to Flash may occur on a microprocessor unit to free up the cache.” *Id.* at 10:55–58. On the

other hand, if the RAM cache is not full (in addition to certain other conditions), “the process ends for that write at step 512 and the data stays in RAM until it is purged from RAM or eventually written to Flash when one or more of the conditional states change.” *Id.* at 11:4–8.

D. Illustrative Claims

Of the challenged claims, claim 1 is the sole independent claim. Claims 2 and 5–8 depend directly from claim 1 and claim 3 depends directly from claim 2. Claim 1 recites:²

- 1[pre] A data storage device comprising:
- 1[a] a host controller interface;
- 1[b] a plurality of microprocessor units each having a portion of random access volatile memory (RAM) dedicated thereto, the RAM portion having a preset threshold for valid data;
- 1[c] a plurality of non-volatile memory device configurations each having dedicated bus connections to individual ones or multiples of the microprocessor units;
- 1[d] a dataflow controller accessible to the host controller interface for managing access to the device configurations;
- 1[e] wherein the dataflow controller periodically compares valid data volume in the RAM portion associated with each microprocessor with the preset threshold, and upon a threshold being met or exceeded, selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory bussed to that microprocessor, and marks locations from which the data is moved as available for writing in the volatile portion.

Ex. 1001, 12:33–53.

² Reference letters in brackets mirror those provided by Petitioner. Pet. 3.

E. Evidence

The Petition relies on the following references:

1. Nakamura et al., US 7,464,221 B2, issued Dec. 9, 2008. (“Nakamura”). Ex. 1004.
2. Mizushima et al., US 7,761,655 B2, issued July 20, 2010. (“Mizushima”). Ex. 1005.
3. Yiming Hu, et al., “RAPID-Cache—A Reliable and Inexpensive Write Cache for High Performance Storage Systems,” 13 IEEE Transactions on Parallel and Distributed Systems 290 (March 2002). (“Hu”). Ex. 1006.
4. Benhase et al., US 7,260,679 B2, issued Aug. 21, 2007. (“Benhase”). Ex. 1007.

Petitioner also relies on the Declaration of Vivek Subramanian, Ph.D. Ex. 1003.

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–3 and 5–8 are unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1–3, 5–8	103	Nakamura, Benhase
1–3, 5–8	103	Mizushima, Hu

III. ANALYSIS OF THE ASSERTED GROUNDS

For the reasons below, we determine Petitioner has not shown a reasonable likelihood of prevailing in its challenges to claims 1–3 and 5–8 of the ’320 patent. Before we analyze the asserted grounds, we first address two matters that will underlie our analysis: the appropriate level of ordinary skill in the art and the constructions we will apply to claim terms.

A. Legal Standards

“In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is

unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)); *see also* 37 C.F.R. § 42.104(b) (requiring a petition for *inter partes* review to identify how the challenged claim is to be construed and where each element of the claim is found in the prior art patents or printed publications relied upon).

A claim is unpatentable as obvious under 35 U.S.C. § 103(a) 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). The question of obviousness is resolved on the basis of underlying factual determinations, 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) where in evidence, so-called “secondary considerations,” including commercial success, long-felt but unsolved needs, failure of others, and unexpected results. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

B. Level of Ordinary Skill in the Art

The level of ordinary skill in the pertinent art at the time of the invention is relevant to how we construe the patent claims. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc). It is also one of the factual considerations relevant to obviousness. *See Graham*, 383 U.S. at 17–18.

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 contends a person of ordinary skill in the art would have had at least bachelor’s degree in electrical engineering or an equivalent, plus three years of experience in the field of memory-device design. Additional education in electrical engineering such as a masters or doctorate degree may serve as a substitute for experience in the field. A person with less education but more relevant practical experience may also meet this standard

Pet. 28 (citing Ex. 1003 ¶¶ 27–28).

Patent Owner does not submit a proposed level of ordinary skill in the art in its Preliminary Response. *See generally* Prelim. Resp. For purposes of this Decision, we adopt Petitioner’s proposed level of ordinary skill.

C. Claim Construction

In this *inter partes* review, the Board applies the same claim construction standard as that applied in federal courts. *See* 37 C.F.R. § 42.100(b). Under this standard, claim terms “are generally given their ordinary and customary meaning” as understood by a person of ordinary skill in the art in question at the time of the invention. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (en banc) (citations omitted). 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. “In determining the meaning of the disputed claim limitation, 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) (citing *Phillips*, 415 F.3d at 1312–17).

Extrinsic evidence is “less significant than the intrinsic record in determining ‘the legally operative meaning of claim language.’” *Phillips*, 415 F.3d at 1317 (citation omitted).

Petitioner asserts “that no express construction of” the terms “non-volatile memory device configurations,” as recited in the challenged claim 1 and “Flash channels,” as recited in claims 2 and 3, “is necessary for purposes of this petition.” Pet. 28–29. Patent Owner agrees with Petitioner that “no express construction of these terms is necessary at this stage” in the proceeding. Prelim. Resp. 6. We determine no terms need to be construed to resolve the disputes between the parties at this stage of the proceeding.

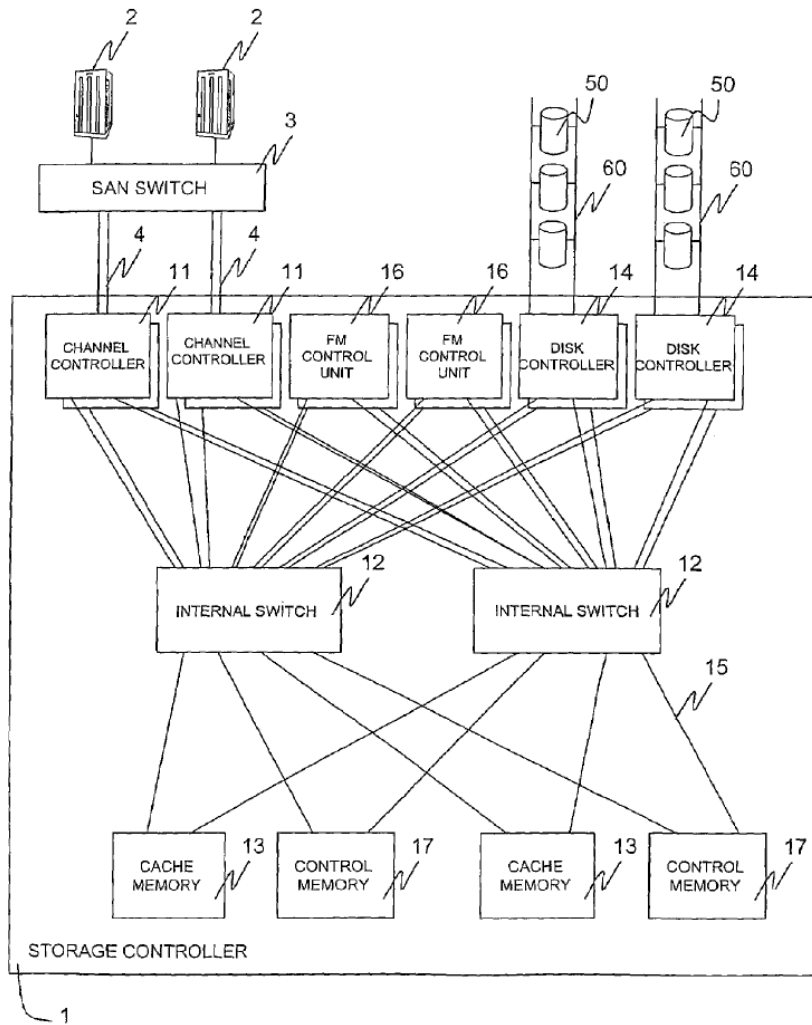
D. Description of Prior Art References

Petitioner’s challenges rely on Nakamura, Benhase, Mizushima, and Hu. *See* Pet., *passim*.

1. Nakamura (Ex. 1004)

Nakamura describes a storage system in which a host computer stores data onto non-volatile storage media, e.g., Flash memory, and a cache memory for temporarily storing data from the host computer. Ex. 1004, code (57); *see id.* at 1:40–42. Figure 1, reproduced below, is a block diagram of one embodiment of Nakamura’s storage system. *Id.* at 2:56–57.

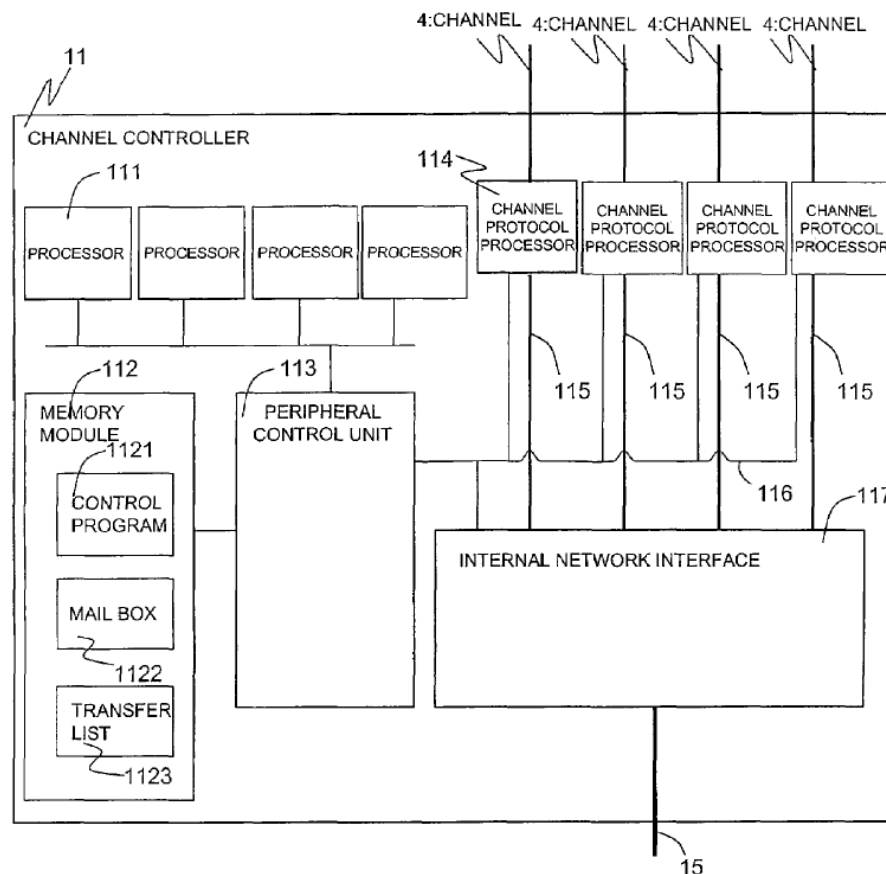
FIG.1



As shown in Figure 1, the storage system includes storage controller 1 and a flash memory device, i.e., “FM control unit 16 internally including flash memory devices.” *Id.* at 3:50–53. Further, “storage controller 1 is connected to a host computer 2.” *Id.* at 3:53–54. Still further, “storage controller 1 includes a plurality of channel controllers 11, a plurality of cache memories 13, a control memory 17, . . . a plurality of FM control units 16, and internal switches 12 connecting these through an internal bus 15.” *Id.* at 3:58–62. When “channel controller 11 receives an input/output request from the host computer 2 through channel 4,” e.g., a write request, the “cache memories temporarily stores [the] data” requested to be stored in

the flash memory. *Id.* at 3:62–4:2; *see id.* 4:11–15, 5:6–10. Figure 2, reproduced below, is a block diagram detailing the configuration of channel controller 11. *Id.* at 2:57–58.

FIG.2

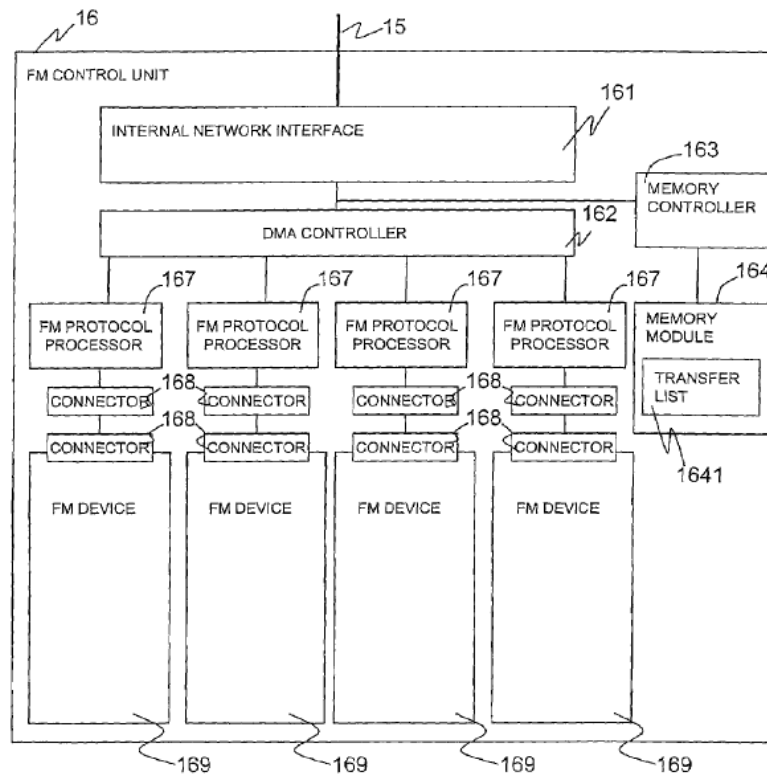


As shown in Figure 2, channel controller 11 includes plurality of processors 111, memory module 112, peripheral control unit 113, and plurality of channel protocol processors 114. *Id.* at 4:26–35. Peripheral control unit 113 is connected to the memory module 112 and controls the memory module. *Id.* at 4:31–32. Processors 111 access the memory module 112 through the peripheral control unit 113, and perform processing based on control program 1121, stored in memory module 112. *Id.* at 4:46–48.

Returning briefly to Figure 1, storage controller 1 includes FM control units 16 which “control of flash memories or the flash memory devices.” *Id.*

at 4:11–12. For example, “FM control units 16 fetch and store the data requested from the host computer 2 to the flash memories or the flash memory 15 device based on a request of the channel controllers 11.” *Id.* at 4:11–15. Figure 4, reproduced below, is a block diagram detailing the configuration of FM control unit 16. *Id.* at 2:21–63.

FIG. 4



Flash memory device 169 of Figure 4 “is used as a memory element.” *Id.* at 6:9–10. In this embodiment of FM control unit 16, flash memory device 169 is “disposed separately” and “connected to the FM control unit 16 through a connector 168.” *Id.* at 6:9–11. In this embodiment of flash memory device 169, “[d]ata is exchanged with the outside using a special protocol. Thus, a FM protocol processor 167 converts the data into a format that can be processed in the storage controller 1.” *Id.* at 6:22–24.

As noted above, host computer 2 may request that data is written to flash memory. *Id.* at 4:11–15, 8:26–29. For the “write request, data is temporarily stored in the cache memory 13 . . . when the [write] target is an flash memory area.” *Id.* at 8:29–31. That write process includes “check[ing] whether there is the data stored in the cache memory 13” and “if there is the data in the read cache area 131, the relevant [cache memory] slot is invalidated.” *Id.* at 8:36–43, 8:60–63. In that case, “the existent data in the cache memory 13 is stored in . . . the flash memory (destaging^[3])” so that “the area [in cache memory] for storing the data from the host computer 2 this time is created.” *Id.* at 8:66–9:2. “[T]he data to be destaged is determined by an LRU (Least Recently Used) algorithm, etc.” *Id.* at 9:3–4. For example, “the earliest accessed [cache memory] slot” is destaged. *Id.* at 12:41–43. The data to be destaged is read “from the cache memory 13” and written “into the flash memory 166 or the flash memory device 169 . . . based on [a] transfer list.” *Id.* at 9:13–17. Subsequently, the requested data to be written from host computer 2 is stored into cache memory 13. *Id.* at 9:25–28; *see id.* at 5:58–63.

Additionally, returning to Figure 1, the storage system further includes hard disk drives (HDDs) 50 connected to storage controller 1. *Id.* at 3:55–63. Data may also be written to HDDs 50, e.g., “processor 111 writes the data in the cache memory 13 into a hard disk drive 50 by a request from the channel controller 11 or at regular time intervals.” *Id.* at 5:28–30.

³ Petitioner explains that “[b]ecause data in a write request is first stored in the write-back cache, the cache inevitably includes data that needs to be written to non-volatile memory—this is called ‘dirty’ data, and the process for writing dirty data to non-volatile memory is called ‘destaging.’” Pet. 15.

2. *Benhase (Ex. 1007)*

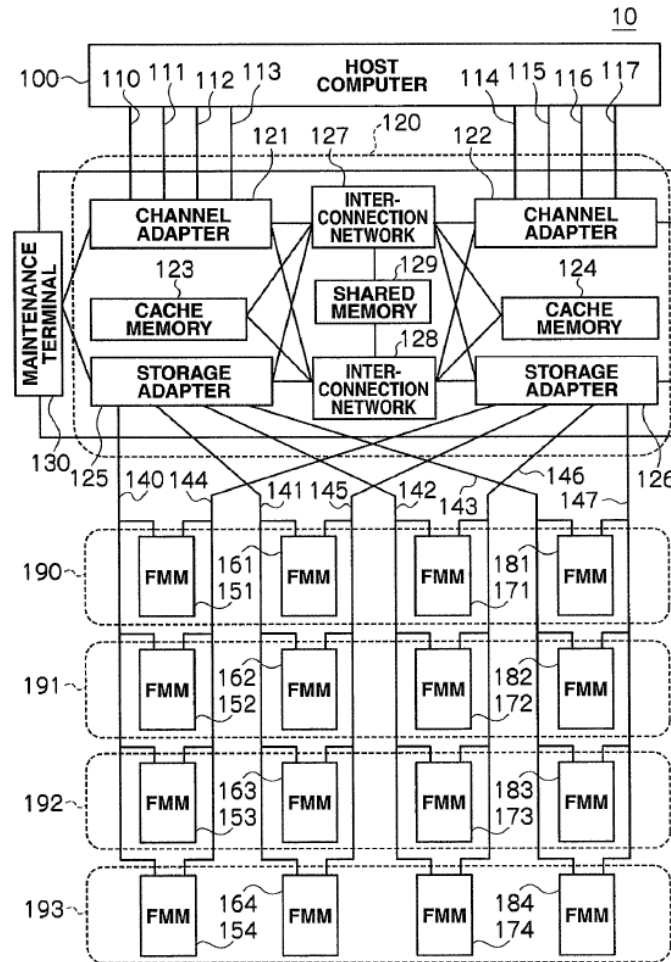
Benhase describes a method to manage a data cache, the data cache “comprising a plurality of tracks, where each track comprises one or more segments.” Ex. 1007, 1:36–39. Benhase aims to provide a “method to manage the tracks residing in the data cache such that tracks having a low reuse potential are preferentially demoted from the cache while tracks having a high reuse potential are preferentially kept in the cache.” *Id.* at 1:28–32. Benhase’s method “maintains a first [least recently used (LRU)] list comprising one or more first tracks having a low reuse potential, maintains a second LRU list comprising one or more second tracks having a high reuse potential and sets a target size for the first LRU list.” *Id.* at 1:39–42; *see id.* at 4:14–15. Benhase “determin[es] if the sum of the number of segments comprising the first LRU list and the second LRU list exceeds a predetermined number, i.e. exceeds a `fixed_predetermined_number`. In certain embodiments, that `fixed_predetermined_number` is set to equal 90 percent of the maximum number of segments that the cache can hold.” *Id.* at 6:2–8. If “the aggregate number of segments comprising the first LRU list and the second LRU list exceed the `fixed_predetermined_number`, and . . . no tracks have been demoted from the cache, then [the] method sets the first LRU list target size to equal the number of segments comprising the first LRU list.” *Id.* at 6:11–17. Using that procedure, Benhase’s method “autonomically adjusts [the] first LRU list target size based upon actual utilization of the tracks in the cache.” *Id.* at 6:18–20.

3. *Mizushima (Ex. 1005)*

Mizushima describes a “storage system [that] includes a flash memory, a cache memory, and a controller for controlling the reading, writing and deletion of data of the flash memory and the reading and writing

of data of the cache memory.” Ex. 1005, code (57). Figure 1, reproduced below, “is a diagram showing a simplified hardware configuration of a storage system 10.” *Id.* at 4:8–10.

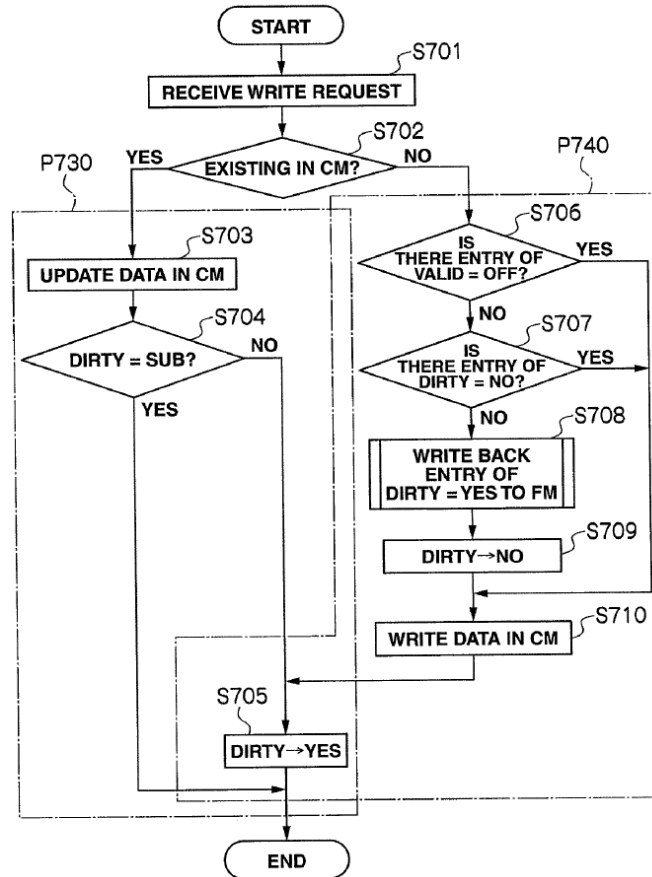
FIG.1



As shown in Figure 1, “storage system 10 comprises a storage controller 120 and flash memory modules (FMM) 151 to 154, 161 to 164, 171 to 174, and 181 to 184.” *Id.* at 4:11–13. “The storage controller 120 comprises channel adapters 121, 122, cache memories 123, 124, storage adapters 125, 126, a shared memory 129, and interconnection networks 127, 128.” *Id.* at 4:13–16. “[C]ache memories 123, 124 are used to temporarily

store [received] data received from the channel adapters 121, 122 or the storage adapters 125, 126, and to permanently store specific received data as needed.” *Id.* at 4:43–46. Further, “cache memories 123, 124, for example, are dynamic random access memories.” *Id.* at 4:46–48. “[S]hared memory 129 stores a table for managing . . . stored data” and “can refer to and set such table.” *Id.* at 4:48–51. Still further, “shared memory 129, for example, is a dynamic random access memory.” *Id.* at 4:52–54.

In addition, as shown in Figure 1, host computer 100 is connected to storage system 100 through the channel adapters. *Id.* at 4:31–34. Host computer 100 “requests the storage controller 120 to read and write data from and in the storage system 10.” *Id.* at 4:34–37. “[S]torage controller 120 uses the channel adapters 121, 122 to interpret the foregoing requests, and uses the storage adapters 125, 126 to read and write data of the flash memory modules 151 to 154, 161 to 164, 171 to 174, 181 to 184 in order to satisfy the requests.” *Id.* at 4:37–42. Figure 12, reproduced below, “is a flowchart showing the processing to be performed by the storage controller 120 and the memory controller 310 regarding the data write request from the host computer 100.” *Id.* at 14:1–4.



First, “storage controller 120 receives the logical page address and new data to be written as the write request (S701), and searches whether an entry including the data of the logical page address exists in the cache memory 123.” *Id.* at 14:5–9. If the entry does not exist, “the storage controller 120 searches whether an entry (empty entry) where the valid flag 5003 is ‘off.’” *Id.* at 14:19–21. “If an empty entry exists (S706: YES), the received new data is written into the relevant entry (S710)” and “the storage controller 120 sets the valid flag 5003 of that entry to ‘on: (in use).’” *Id.* at 14:21–25. But, “if an empty entry does not exist (S706: NO), the storage controller 120 searches whether an entry (original data retention entry) where the dirty register 5005 is ‘no’ exists (S707).” *Id.* at 26–29. Then, “[i]f an original data retention entry does not exist (S707:NO), the storage controller 120 performs write-back of the stored data of the entry (update

data retention entry) where the dirty register 5005 is ‘yes’ to the flash memory 320” and the new data is written into the relevant entry in step S710. *Id.* at 14:33–40. An LRU algorithm is used for selecting the entry to be subject to the write-back processing. *Id.* at 14:40–42.

4. *Hu (Ex. 1006)*

Hu describes a “a new cache architecture called RAPID-Cache for Redundant, Asymmetrically Parallel, and Inexpensive Disk Cache” which “consists of two redundant write buffers on top of a disk system.” Ex. 1006, 290. Hu further explains that “[i]n a traditional RAID [disk] system with a write cache, dirty data in the write cache is written into the RAID system in a process called destage.” *Id.* at 294; *see id.* at 291. Hu’s “RAPID-Cache also requires destaging.” *Id.* at 294. In Hu, “one or several destaging threads are initiated when the controller detects an idle period, or when the number of dirty blocks in the primary write cache exceeds a high water-mark, say 70 percent of the cache capacity.” *Id.* “[D]estaging threads find a dirty LRU block in the primary cache, read the old data and parity of that block from disks or the read cache, compute the new parity, and write the new data and parity to disks.” *Id.* “The destaging threads run continuously until the idle period is over, or until the dirty block count in the primary cache falls below a low water-mark, say 30 percent of the cache capacity.” *Id.*

E. Obviousness over Nakamura and Benhase (Ground 1) and Mizushima and Hu (Ground 2)

Petitioner argues that claims 1–3 and 5–8 of the ’320 patent would have been obvious over the combination of Nakamura and Benhase (Ground 1) and also over the combination of Mizushima and Hu (Ground 2). Pet. 6. Patent Owner opposes Petitioner’s contentions. Prelim. Resp. 16–

36. Based on our review of the record before us, we determine that Petitioner has not demonstrated a reasonable likelihood that it would prevail in showing that claims 1–3 and 5–8 are unpatentable as obvious over either the combination of Nakamura and Benhase or the combination of Mizushima and Hu.

1. *1[e] wherein the dataflow controller . . . selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory*

A central point of dispute between the parties is whether the prior art references teach a dataflow controller that “selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory.” For both grounds, Petitioner relies on references that disclose a Least Recently Used (LRU) algorithm as teaching this disputed limitation. For example, with respect to Ground 1, Petitioner relies on Nakamura and argues its channel controllers 11, select data to move from cache memory 13 to an FM device 169 using an LRU algorithm. Pet. 57 (citing Ex. 1004, 9:3–4, 11:65–12:3, 12:18–24, Fig. 12A; Ex. 1003 ¶ 141). According to Petitioner, Nakamura’s “channel controllers 11 cycle through the dedicated cache for each FM device and select the ‘earliest accessed slot’ for destaging—i.e., the slot going *longest* without being written to in the portion of cache assigned to a Flash memory device.” Pet. 57 (citing Ex. 1004, 12:41–43, Fig. 10C, 10:66–11:3, 11:23–27; Ex. 1003 ¶ 141).

With respect to Ground 2, Petitioner argues Mizushima selects the data to move from cache memory to flash memory modules using a traditional LRU algorithm, which, according to Petitioner, selects data that

has been longest in the cache for writing back to flash. Pet. 77 (citing Ex. 1005, 14:40–42, 1:60–67; Ex. 1003 ¶ 194.)

Patent Owner argues the phrase “selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory” refers to a first-in-first-out (FIFO) algorithm not an LRU algorithm. Prelim. Resp. 18. Patent Owner argues that the LRU algorithms relied upon by Petitioner in Nakamura and Mizushima are different than the FIFO algorithm recited in claim 1 because an “LRU policy is concerned with the time that data was last accessed (e.g. **read or written**), while a FIFO policy is concerned with the amount of time the data has been present in the cache.” Prelim. Resp. 19, *see also id.* at 17, 32.

In its Reply Brief, Petitioner responds that Patent Owner’s interpretation of the claim as reciting a FIFO algorithm ignores the “without alteration” portion of the disputed claim limitation. Reply Br. 1. That phrase, according to Petitioner, makes clear that the controller selects and moves data based not only on how long the data has been in RAM, but also on whether that data has been altered. Reply Br. 1–2. Thus, Petitioner argues that Patent Owner’s interpretation leads to a policy that moves data based solely on length of time the data has been in RAM without considering whether the data has been altered. *Id.* at 2. Petitioner argues that the Specification of U.S. Patent No. 7,464,240 (the ’240 patent), which is incorporated by reference in the ’320 patent, does not support interpreting the claim as requiring a FIFO algorithm and instead describes an algorithm where the oldest data is not destaged if it has been altered. Reply Br. 2–3 (citing Ex. 1018, 12:6–10, 11:7–14, Fig. 5. Petitioner also argues that Patent Owner’s interpretation of the claim contradicts its interpretation of the same

limitation in the related IPR2022-00115 proceeding. Reply Br. 3–4 (citing *Hewlett Packard Enterprise Co. v. Intellectual Ventures II LLC*, IPR2022-00115, Paper 8 at 6). Finally, Petitioner argues that extrinsic evidence demonstrates that LRU considers whether data in cache has been altered (Ex. 2001, 2, 3) while FIFO is just concerned with the amount of time the data has been present in the cache. *Id.* at 4 (citing Ex. 2001, 2, 3).

In its Sur-reply Patent Owner makes two primary arguments. First, Patent Owner argues that the phrase “selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory” refers to a FIFO algorithm because when old data is altered, it becomes new data and thus data that is longest in volatile memory is necessarily longest without alteration. Sur-reply 1. Second, Patent Owner argues that even if “without alteration” is interpreted in the way Petitioner contends (i.e. even if claim 1 does not recite a FIFO algorithm), an LRU algorithm still does not meet the claim language because the LRU algorithm “looks at any access, including *read accesses*, not just the time since data was last written.” Sur-reply 2 (citing Ex. 1004, 9:2–4; Ex. 1007, 2:39–44, 4:14–15). A read access, according to Patent Owner, does not alter data and thus an LRU algorithm does not satisfy the claim language. *Id.* at 2.

We are unpersuaded by Petitioner’s arguments. Petitioner does not sufficiently explain how the LRU algorithms disclosed in Nakamura and Mizushima, which would move data longest in memory without being *accessed*, teaches moving data longest in memory without *alteration*.⁴

⁴ Because we determine that Petitioner does not sufficiently demonstrate that an LRU algorithm teaches the disputed limitation, we need not make any

Petitioner acknowledges that in the LRU algorithm a determination is made as to which data has been in memory longest without being either read or written to (i.e. accessed). Pet. 16 (explaining that the LRU algorithm “determines which data has been in cache longest without being used (e.g. read or overwritten) and preferentially destages that data over other data.”) Nakamura confirms that the LRU algorithm is concerned with when data was last accessed and that these accesses include both reading and writing data. *See* Ex. 1004, 11:65–12:2 (“Assuming the use of the LRU algorithm, the list stores information including what number is which slot recently accessed.”); 11:56–59 (“The number of access-times field 1064 indicates the number of accesses including reading and writing occurred on the medium area.”)

We, therefore, determine that one of ordinary skill in the art would understand that the LRU algorithms disclosed in Nakamura and Mizushima are concerned both with data being read as well as data being written. We also determine that one of ordinary skill would understand, as Patent Owner argues, that a read access does not alter data. Sur-reply 2. Petitioner’s arguments do not sufficiently address how the LRU algorithm, which moves data that has been in memory longest without being read or written, teaches moving data that has been longest in the volatile memory portion without alteration.

For example, Petitioner argues that Nakamura’s “channel controllers . . . select the ‘earliest accessed slot’ for destaging—i.e., the slot going ***longest*** without being written to” (Pet. 57) but does not address or account

determination as to whether the disputed limitation is directed to a FIFO algorithm.

for Nakamura’s selection process when data has been read. Similarly, Petitioner argues that Mizushima uses a “traditional LRU algorithm” but does not sufficiently address Mizushima’s selection process for a read access.⁵ Dr. Subramanian’s testimony is nearly identical to Petitioner’s contentions set forth in the Petition without additional citations to evidence or explanation and thus does not provide any additional support. *See* Ex. 1003 ¶¶ 141, 194. Thus, based on the arguments and evidence presented at this stage of the proceeding, we determine Petitioner does not sufficiently demonstrate that an LRU algorithm teaches the disputed claim limitation.

2. Conclusion

For the foregoing reasons, we are not persuaded that the cited evidence adequately supports Petitioner’s contention that the combination of Nakamura and Benhase (Ground 1) and the combination of Mizushima and Hu (Ground 2) teach the limitation “selects and moves data that has been longest in the volatile memory portion without alteration from the RAM portion to the non-volatile memory” of claim 1. Because all challenged claims depend directly or indirectly from claim 1, we determine Petitioner has not demonstrated a reasonable likelihood of success in establishing the unpatentability of any challenged claims over the combination of Nakamura and Benhase or the combination of Mizushima and Hu.

⁵ In arguing that Mizushima uses a traditional LRU algorithm, Petitioner cites a statement from the Background section of Mizushima which describes the destaging process as migrating a data file “in which the longest time has lapsed from the last writing.” Ex. 1005, 1:60–64. Petitioner does not explain, however, whether this statement, which only mentions write accesses rather than read accesses, overrides their explicit reliance on a traditional LRU algorithm for selecting data to be migrated. *See* Ex. 1005 14:40–42.

IV. CONCLUSION

For the foregoing reasons and on the present record, we determine that the Petition does not demonstrate a reasonable likelihood that Petitioner would prevail in establishing the unpatentability of at least one of the challenged claims of the '320 patent.

V. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that pursuant to 35 U.S.C. § 314(a), an *inter partes* review is not instituted on the challenged claims of the '320 patent.

IPR2022-00116
Patent 7,882,320 B2

For PETITIONER:

Eliot Williams
Lauren Dreyer
Lindsay Volpenhein Cutie
Jacob McDonald
BAKER BOTTS LLP
eliot.williams@bakerbotts.com
lauren.dreyer@bakerbotts.com
lindsay.volpenhein@bakerbotts.com
jacob.mcdonald@bakerbotts.com

For PATENT OWNER:

Daniel S. Block
Byron L. Pickard
James R. Hietala
Steven M. Pappas
Todd C. Thurheimer
STERNE, KESSLER, GOLDSTEIN & FOX P.L.L.C
dblock-PTAB@sternekessler.com
bpickard-PTAB@sternekessler.com
JHietala-PTAB@sternekessler.com
spappas-PTAB@sternekessler.com
TThurheimer-PTAB@sternekessler.com

Russel J. Rigby
INTELLECTUAL VENTURES
rrigby@intven.com