

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION, LENOVO (UNITED STATES) INC., and
NETAPP, INC.,
Petitioner,

v.

INTELLECTUAL VENTURES I LLC,
Patent Owner.

Case IPR2017-00429
Patent 6,775,745 B1

Before JEFFREY S. SMITH, GEORGIANNA BRADEN, and
DANIEL J. GALLIGAN, *Administrative Patent Judges*.

SMITH, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision issues pursuant to 35 U.S.C. § 318(a). For the reasons that follow, we determine Petitioner has shown by a preponderance of the evidence that claims 4–6 of U.S. Patent No. 6,775,745 B1 (Ex. 1001, “the ’745 patent”) are unpatentable. Additionally, we dismiss as moot the Parties’ Motions to Exclude.

A. Procedural History

EMC Corporation, Lenovo (United States) Inc., and NetApp, Inc. (collectively “Petitioner”) filed a Petition for *inter partes* review of claims 4, 5, and 6 of the ’745 patent. Paper 5 (“Pet.”). Intellectual Ventures I LLC (“Patent Owner”) filed a Preliminary Response. Paper 9 (“Prelim. Resp.”). Pursuant to 35 U.S.C. § 314(a), the Board instituted an *inter partes* review of claims 4 and 5 as unpatentable under 35 U.S.C. § 103(a) over Lee,¹ Dharap,² and Karedla.³ See Paper 11, 29 (“Dec. on Inst.”).

After institution of trial, Patent Owner filed a Patent Owner Response (Paper 21, “PO Resp.”), to which Petitioner filed a Reply (Paper 26, “Reply”). In addition, Patent Owner filed objections to Evidence in Petitioner’s Reply (Paper 27), a List of New Arguments in the Reply (Paper

¹ Donghee Lee et al., *Implementation and Performance Evaluation of the LRFU Replacement Policy*, IEEE (1997) (“Lee,” Ex. 1003).

² Chanda Dharap, U.S. Patent Application Publication No. US2002/0078300 A1 (published June 20, 2002) (“Dharap,” Ex. 1004).

³ Ramakrishna Karedla et al., *Caching Strategies to Improve Disk System Performance*, COMPUTER, 27(3):38–46 (March 1994) (“Karedla,” Ex. 1006).

30), and a Motion for Observations (Paper 32). Petitioner responded to the Motion for Observations (Paper 37).

An oral argument was held on January 4, 2018. A transcript of the oral argument is included in the record. Paper 41 (“Tr.”).

On April 24, 2018, the Supreme Court issued its decision in *SAS Institute Inc. v. Iancu*, 138 S.Ct. 1348 (2018). On May 7, 2018, the Board modified the institution decision to include review of all challenged claims and all grounds presented in the Petition, including an *inter partes* review of (1) claims 4–6 as unpatentable under 35 U.S.C. § 103(a) based on Burton⁴ and Karedla; (2) claims 4–6 as unpatentable under 35 U.S.C. § 103(a) based on Burton and Sweeney⁵; (3) claims 4 and 5 as unpatentable under 35 U.S.C. § 103(a) based on Lee, Dharap, and Karedla; (4) claim 6 as unpatentable under 35 U.S.C. § 103(a) based on Lee, Dharap, Karedla, and Robinson⁶; (5) claims 4 and 5 as unpatentable under 35 U.S.C. § 103(a) based on Lee, Dharap, and Sweeney; and (6) claim 6 as unpatentable under 35 U.S.C. § 103(a) based on Lee, Dharap, Sweeney, and Robinson.

⁴ David Alan Burton & Erez Webman, U.S. Patent No. 6,738,865 B1 (issued May 18, 2004) (“Burton,” Ex. 1005).

⁵ Adam Sweeney et al., Scalability in the XFS File System, 1996 USENIX TECHNICAL CONFERENCE (Jan. 22–26, 1996) (“Sweeney,” Ex. 1012).

⁶ As presented in the Petition, “Robinson” refers collectively to two references: (1) John T. Robinson & Murthy V. Devarakonda, *Data Cache Management Using Frequency-Based Replacement*, PERFORMANCE EVALUATION REV., 18(1):134–142 (May 1990) (“the Robinson article”), Ex. 1007, and (2) John T. Robinson, U.S. Patent No. 5,043,885 (Aug. 27, 1991) (“Robinson ’885,” Ex. 1008).

Patent Owner filed a Supplemental Response (Paper 53, “PO Supp. Resp.”). Petitioner responded by filing a Supplemental Reply (Paper 56, “Pet. Supp. Reply”).

A second oral argument was held on September 7, 2018. *See* Paper 70 (“Supp. Tr.”).

B. Related Matters

The parties identify the following district court cases in which the ’745 patent has been asserted: *Intellectual Ventures I LLC v. NetApp, Inc.*, No. 1:16-cv-10868-IT (D. Mass); and *Intellectual Ventures I LLC v. Lenovo Grp. Ltd.*, No. 1:16-cv-10860-IT (D. Mass). Pet. 3; Paper 6, 1. The ’745 patent is also the subject of an *inter partes* review designated IPR2016-01643.

C. The ’745 Patent

The ’745 patent is directed to a hybrid data caching mechanism. Ex. 1001, 1:6–10. The ’745 patent explains as background that disk input/output (I/O) can take significant amounts of time while searching for and loading programs and data. *Id.* at 1:17–20. Accessing a file requires first accessing the directory, which keeps track of file locations, and then accessing the file itself. *Id.* at 1:28–31. “Multiple seeks are typically required to locate a file in the directory, especially if the file is not in the first part of the root directory.” *Id.* at 1:31–33. Also,

[w]hen more than one program is requesting data, the file system driver can end up reading a small amount of data for a first program, then seek to a different area on the disk to read another small amount for a second program, then seek back to the

original area to read the next small amount of data for the first program, and so forth.

Id. at 1:42–48. To circumvent these problems, file system drivers do a certain amount of caching. *Id.* at 2:1–3. For example, “caching programs strictly use the MRU-LRU (most recently used-least recently used) mechanism as their sole means of deciding what data to keep and what data to discard.” *Id.* at 2:14–17. In using the MRU-LRU approach, “an important file that has not been used recently will be discarded when the cache reaches a target capacity and is forced to free up additional capacity.” *Id.* at 2:17–20. “Strictly adhering to the MRU-LRU mechanism fails to ensure that files which have been used often, but may not have been used recently, are maintained in the cache.” *Id.* at 2:20–23.

The ’745 patent discloses a caching mechanism that “minimizes seeks and reads to a hard drive and which keeps data in the cache based upon currency of use and the number of times the data is used.” *Id.* at 2:31–35. The patent states “system performance is enhanced through the hybrid caching mechanism where a file that has been used often but not recently is maintained in the cache.” *Id.* at 3:48–50.

Specifically, the ’745 patent discloses identifying “a least frequently and least recently used file” and eliminating that file to liberate capacity of the cache. *Id.* at 2:50–53. Figure 2A of the ’745 patent is shown below:

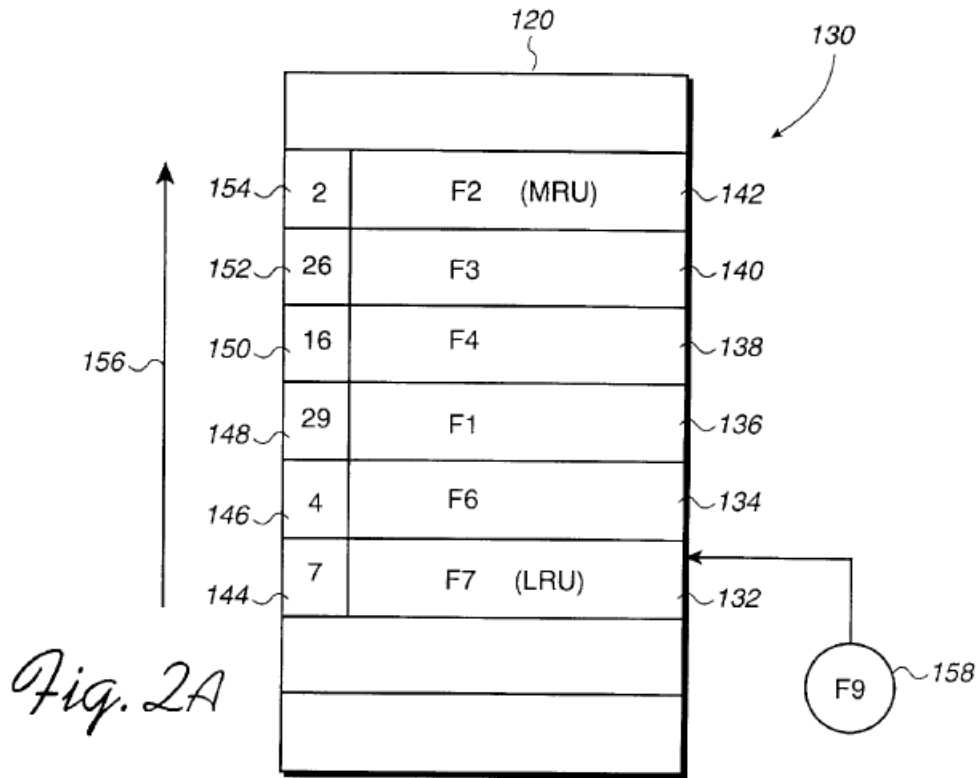


Figure 2A above depicts a structure of a cache according to an embodiment of the invention. *Id.* at 5:53–55. Cache 120 includes files F1 through F7. *Id.* at 5:55–56. File F7 132 is the least recently used (LRU) file. *Id.* at 5:56–57. In other words, file F7 132 has gone the longest time as compared to the other files in the cache without being used by the operating system. *Id.* at 5:57–60. File F2 142 is the most recently used (MRU) file. *Id.* at 5:63. A frequency factor is assigned to each file in the cache. *Id.* at 5:65–66. For example, as illustrated above, file F7 has a frequency factor of seven, file F6 has a frequency factor of four, file F1 has a frequency factor of twenty-nine, and so on. *Id.* at 6:4–8. In one embodiment, the frequency factor is increased with each use of the file; in another embodiment, the frequency factor is decreased with the lack of use of the corresponding file. *Id.* at 5:66–6:4.

The '745 patent discloses that once the target capacity of the cache is reached, the frequency factors are scanned, beginning (in one embodiment) with the frequency factor for the LRU file and ending with the MRU file. *Id.* at 6:8–12. The scanning of the frequency factors results in “a least frequently and least recently used file” being identified. *Id.* at 2:50–51, 12:41–44, 13:1. In the same embodiment, the scanning of the frequency factors “determines which file . . . is the least frequently used (LFU) file which has been least recently used.” *Id.* at 6:18–21. The '745 patent discloses an embodiment in which “the frequency factor for the MRU file is not considered when determining the LFU file since the MRU file has been so recently placed in the cache and has not had an opportunity to be reused.” *Id.* at 6:39–43. In the example illustrated in Figure 2A above, the scan will find file F6 134 as the LRU file with the lowest frequency factor—i.e., the LFU file that has been least recently used. *Id.* at 6:46–49. A new file to be placed in the cache, such as file F9 158, will replace file F6 134. *Id.* at 6:52–55.

In one embodiment, the caching mechanism described above is self-tuning. *Id.* at 6:56–57. For example, after a user finishes watching a movie on the system and then performs word processing operations, the cache will have frequently used movie files that will not be used again. *Id.* at 6:57–63. The caching mechanism will decrement the frequency factor associated with the files by a factor of one or greater over an arbitrary period so that the files are removed from the cache. *Id.* at 6:53–7:3. Other tuning features include how often to decrement the frequency factors, how far back to search from the LRU file, how far forward to search towards the LRU file, how much to decrement the factors by, and how to assign weight to certain files with

frequency factors upon their initial read into the cache to guarantee the certain files remain above others. *Id.* at 7:3–15.

The self-tuning aspects also include embodiments to adjust the frequency factors if “the least frequently and least recently used file” is in a defined proximity to the MRU file so that a file other than the MRU is identified as the least frequently and least recently used file. *Id.* at 7:16–24. In one embodiment, only the frequency factors associated with the files proximate to the LRU are decremented. *Id.* at 7:24–35. In one embodiment, if the least frequently and least recently used file is in the top 1/3 i.e., the 1/3 of the cache unit logically proximate to the MRU file, then “the least frequently and least recently used file” is not chosen from the top 1/3 of the cache unit. *Id.* at 7:36–43. In one embodiment, the frequency factors associated with files in the bottom 1/3 of the unit are decremented until a “least frequently and least recently used file” is identified in the bottom 1/3 of the cache unit. *Id.* at 7:43–47. “In one embodiment, the user is provided an application to customize the tuning to their particular situation.” *Id.* at 7:11–13. The self-tuning features can be optimized by the user through a graphical user interface. *Id.* at 7:49–51.

The ’745 patent describes implementing the improved caching mechanism “in conjunction with reading extended segments of data.” *Id.* at 4:32–36. The ’745 patent states that “large reads (greater than 64 Kbyte) eliminate the overhead, i.e., seek times, rotational latencies, transfer times, etc., associated with performing two, four or more reads at 32 Kbytes or less as performed in the prior art.” *Id.* at 5:38–42. For example, the ’745 patent describes that “when the first block of the file is read[,] an additional 64, 128, or 256 Kbytes of data are read with it.” *Id.* at 11:22–24. The ’745

patent states that “the reads of the extended segments allows for the minimization of seeks and reads from the storage medium, since the files are transferred to the cache upon the initial read.” *Id.* at 11:45–48.

D. Illustrative Claim

Among the challenged claims (claims 4–6), claim 4 is independent. Claim 4 is illustrative of the subject matter of the challenged claims and is reproduced as follows:

4. A caching method for enhancing system performance of a computer, comprising:

reading an extended segment of data in response to a request from an operating system;

storing copies of files associated with the extended segment in a cache;

assigning frequency factors to each of the files stored in the cache, the frequency factors indicating how often each of the corresponding files are requested by the operating system;

scanning the frequency factors, the scanning being performed in response to a target capacity of the cache being attained;

identifying a least frequently and least recently used file;
and

eliminating the least frequently and least recently used file to liberate capacity of the cache.

Id. at 12:54–13:4.

II. DISCUSSION

A. *Level of Ordinary Skill in the Art*

In determining whether an invention would have been obvious at the time it was made, we consider the level of ordinary skill in the pertinent art at the time of the invention. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17 (1966). “The importance of resolving the level of ordinary skill in the art lies in the necessity of maintaining objectivity in the obviousness inquiry.” *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991). The person of ordinary skill in the art is a hypothetical person who is presumed to have known the relevant art at the time of the invention. *In re GPAC, Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995). The level of ordinary skill in the art may be reflected by the prior art of record. *Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001). Factors that may be considered in determining the level of ordinary skill in the art include, but are not limited to, the types of problems encountered in the art, the sophistication of the technology, and educational level of active workers in the field. *GPAC*, 57 F.3d at 1579. In a given case, one or more factors may predominate. *Id.* Generally, it is easier to establish obviousness under a higher level of ordinary skill in the art. *Innovention Toys, LLC v. MGA Entm’t, Inc.*, 637 F.3d 1314, 1323 (Fed. Cir. 2011) (“A less sophisticated level of skill generally favors a determination of nonobviousness . . . while a higher level of skill favors the reverse.”).

Petitioner’s declarant, Dr. John Kubiadowicz (“Dr. Kubiadowicz”), testifies that a person of ordinary skill in the art at the relevant time “would have had at least a bachelor’s degree in computer science or electrical engineering and 3-5 years of professional experience in computer systems

design, or a master's or doctorate and 1-2 years of professional experience in computer systems design, or equivalent academic experience.” Ex. 1002 ¶ 23. Patent Owner's declarant, Dr. Frederic T. Chong (“Dr. Chong”), testifies “while I agree with Dr. Kubiawicz that a person of ordinary skill in the art would have a bachelor's, master's, or doctorate degree in computer science or electrical engineering, I believe that the person of ordinary skill in the art with a bachelor's degree would have 1-2 years of professional experience and that a person of ordinary skill in the art with a master's degree or doctorate would not necessarily have any professional experience.” Ex. 2004 ¶ 23. Based on the evidence of record, including the testimony of the parties' declarants, the subject matter at issue, and the prior art of record, we determine that Patent Owner's proposed skill level is appropriate, and we adopt Patent Owner's articulation of the level of ordinary skill in the art for our analysis below. Ex. 2004 ¶ 23.

B. Principles of Law

A claim is unpatentable under 35 U.S.C. § 103(a) if “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person 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) objective evidence of nonobviousness,

i.e., secondary considerations, if in evidence. *See Graham*, 383 U.S. at 17–18.

“A determination of whether a patent claim is invalid as obvious under § 103 requires consideration of all four *Graham* factors, and it is error to reach a conclusion of obviousness until all those factors are considered.” *Apple v. Samsung Elecs. Co., Ltd.*, 839 F.3d 1034, 1048 (Fed. Cir. 2016) (en banc) (citations omitted). “This requirement is in recognition of the fact that each of the *Graham* factors helps inform the ultimate obviousness determination.” *Id.*

“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”)). This burden never shifts to Patent Owner. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (citing *Tech. Licensing Corp. v. Videotek, Inc.*, 545 F.3d 1316, 1326–27 (Fed. Cir. 2008)) (discussing the burden of proof in *inter partes* review). Furthermore, Petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016).

Thus, to prevail in an *inter partes* review, Petitioner must explain how the asserted prior art or the proposed combinations of prior art would have rendered the challenged claims unpatentable. At this final stage, we determine whether a preponderance of the evidence of record shows that the challenged claims would have been obvious over the proposed combinations

of prior art. We analyze the instituted grounds of unpatentability in accordance with the above-stated principles.

C. Claim Construction

In an *inter partes* review, we construe claim terms in an unexpired patent according to their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b) (2016); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2144–46 (2016) (upholding the use of the broadest reasonable interpretation standard). Consistent with the broadest reasonable construction, claim terms are presumed to have their ordinary and customary meaning as understood by a person of ordinary skill in the art in the context of the entire patent disclosure. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). In addition, the Board may not “construe claims during [an *inter partes* review] so broadly that its constructions are *unreasonable* under general claim construction principles.” *Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1298 (Fed. Cir. 2015) (emphasis omitted), *overruled on other grounds by Aqua Prods., Inc. v. Matal*, 872 F.3d 1290 (Fed. Cir. 2017). An inventor may provide a meaning for a term that is different from its ordinary meaning by defining the term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

Petitioner proposes constructions for the following terms/phrases: an “extended segment of data” and “identifying a least frequently and least recently used file.” Pet. 15–18. Patent Owner proposes constructions for the phrases “frequency factors indicating how often each of the corresponding

files are requested by the operating system” and “a least frequently and least recently used file.” PO Resp. 7–36; PO Supp. Resp. 3–4. Patent Owner also proposes a construction for “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file.” PO Supp. Resp. 4–6. In its Supplemental Reply, Petitioner disagrees with Patent Owner’s construction of “frequency factors indicating how often each of the corresponding files are requested by the operating system,” “a least frequently and least recently used file,” and “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file.” Pet. Supp. Reply 5–14. For our analysis in this Decision, we determine that only “frequency factor,” “a least frequently and least recently used file,” and “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file” require express construction.

1. “*frequency factor*”

The term “frequency factor” (or “frequency factors”) is recited, directly or indirectly, in each of challenged claims 4–6. In the Final Written Decision in related IPR2016-01643, the Board determined that the broadest reasonable interpretation of “frequency factor” recited in claim 4 of the ’745 patent, read in light of the specification, including the self-tuning embodiments, encompasses at least “an indicator based on frequency of use or access.” IPR2016-01643, Paper 51, 13–18.

Patent Owner contends the broadest reasonable interpretation of the “frequency factors” limitation of claim 4 is “frequency factors *showing, with a fair degree of certainty*, how often each of the corresponding files are requested by the operating system.” PO Resp. 8. In contrast, the

specification of the '745 patent describes self-tuning embodiments where frequency factors are based in part on frequency of use, and based in part on recency of use. See Ex. 1001, 6:2–4 (“the frequency factor is decreased with the lack of use of the corresponding file”), 6:56–7:51, 8:67–9:5, 12:52–53 (“decrementing a frequency factor after a time period of non-use of a corresponding file”). Patent Owner’s declarant, Dr. Chong, provides testimony showing that a frequency factor based in part on frequency of use and based in part on recency of use may not show, with a fair degree of certainty, how often each of the corresponding files were accessed. See Ex. 1026, 53:1–21 (“Q. And it [the frequency factor] can go - -start at a hundred and decrement all the way down to two. That’s fair, right? A. Potentially.”).

Patent Owner contends that it does not matter whether the specification of the '745 patent includes embodiments in which disclosed “frequency factors” are heavily weighted such that they do not “indicat[e] how often each of the corresponding files are requested by the operating system.” PO Supp. Resp. 10. According to Patent Owner, the narrower claim language excludes from the scope of claim 4 any embodiments in which the frequency factors do not indicate how often each file is requested by the operating system. PO Supp. Resp. 11–12 (citing Ex. 2021 ¶¶ 159–160). Patent Owner contends “common sense dictates that a number that purports to be an *indicator* of some quantity needs to at least bear some reasonably reliable relationship to the quantity purportedly being indicated.” *Id.* at 12.

In contrast, Petitioner persuasively argues that the claims themselves contradict Patent Owner’s contention that the claim language excludes the self-tuning embodiments disclosed in the specification. See Pet. Supp.

Reply 5–8. Claim 7, which depends from claim 4, expressly recites that the caching methods are “self-tuning,” and, therefore, the scope of independent claim 4 necessarily encompasses the self-tuning embodiments disclosed in the specification. *See Tr. of Columbia Univ. v. Symantec Corp.*, 811 F.3d 1359, 1370 (Fed. Cir. 2016) (“construing the independent claim to exclude material covered by the dependent claim would be inconsistent.”).

Patent Owner’s construction of “frequency factors” excludes embodiments not only disclosed in the specification but also expressly claimed, which “is rarely, if ever, correct and would require highly persuasive evidentiary support.” *Adams Respiratory Therapeutics, Inc. v. Perrigo Co.*, 616 F.3d 1283, 1290 (Fed. Cir. 2010) (citations omitted). We agree with Petitioner that the intrinsic evidence in the claims and the rest of the specification is more instructive than extrinsic evidence such as a dictionary definition of “indicate.” *See Phillips* at 415 F.3d 1317.

Therefore, we adopt our previous determination that the broadest reasonable interpretation of “frequency factor” recited in claim 4, read in light of the specification, including the self-tuning embodiments, encompasses at least “an indicator based on frequency of use or access.” *See IPR2016-01643 Paper 51, 13–18.*

2. “a least frequently and least recently used file”

Patent Owner contends that the broadest reasonable interpretation of “a least frequently and least recently used file” excludes the most recently used file. PO Supp. Resp. 3. According to Patent Owner, excluding the MRU file is consistent with every embodiment disclosed in the specification

of the '745 patent. *Id.* at 4 (citing Ex. 1001, 6:46–49, 6:51–52, 7:60–8:3, 11:3–5).

Petitioner persuasively argues that the proper construction of this term encompasses an embodiment in the specification of the '745 patent in which the MRU can be identified and eliminated as a least frequently and least recently used file. Pet. Reply 10–11 (citing Ex. 1026, 103:19–104:6 (“Q. But Doctor . . . [t]here’s at least one embodiment [Ex. 1001, 6:19–21], you would concede, that the MRU is an eviction candidate. A. Yes”); Ex. 1001, 6:19–21 (“the scanning of the frequency factors determines which file from the LRU file to the MRU file is the least frequently used (LFU) file which has been least recently used”)).

We determine the broadest reasonable interpretation of “a least frequently and least recently used file,” read in light of the specification of the '745 patent, encompasses the MRU file.

3. *“scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file”*

Patent Owner provides four arguments to support its contention that “*scanning from a LRU file to a MRU file*” specifies that the scanning process “must scan all files from the LRU file to the MRU file, inclusive.”

PO Supp. Resp. 4–6 (citing Ex. 2021 ¶ 166). First, Patent Owner contends that a person of ordinary skill would understand that a phrase having the form “[verb] from [A] to [Z]” expresses that “an action is to be performed on all items within an expressed range, inclusive.” *Id.* at 5 (citing Ex. 2021 ¶ 167). Second, Patent Owner contends that the claim language recites specific end points of a range joined with “from” and “to,” which requires scanning all frequency factors within the range, including the end points. *Id.* (citing Ex. 2021 ¶ 168). Third, Patent Owner contends that claim 15

supports this construction by explicitly excluding the MRU file from the scanning. *Id.* at 6 (citing Ex. 2021 ¶ 169). Fourth, Patent Owner contends the specification of the '745 patent supports this construction by disclosing that the LRU file and the MRU file define the end points of the range of files included in the scanning process. *Id.* (citing Ex. 1001, 6:14–17, 6:49–50; Ex. 2021 ¶ 170).

Petitioner contends that the broadest reasonable construction of the “scanning” limitation, read in light of the specification of the '745 patent, includes “embodiments in which some files in the cache are not scanned, so long as the scanning direction proceeds from the LRU file towards the MRU file, rather than in the opposite direction.” Pet. Supp. Reply 10 (emphasis omitted). Petitioner persuasively contends that the specification discloses “features which may enhance tuning,” such as a user customizing “*how far forward to search from the LRU file.*” Pet. Supp. Reply 10–12 (citing Ex. 1001, 7:3–6; Ex. 1028 ¶¶ 93–94). We agree that the scope of claim 6, read in light of the specification’s disclosure of an embodiment that limits “how far forward to search from the LRU file,” encompasses not scanning some files in the cache, so long as the direction of scanning proceeds “from the LRU file to the MRU file,” but not in the opposite direction.

For the foregoing reasons, we determine that the broadest reasonable interpretation of “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file,” read in light of the specification, encompasses at least “scanning in a direction that proceeds from the LRU file towards the MRU file.”

D. Asserted Obviousness over Burton and Karedla

Petitioner contends that claims 4–6 are unpatentable under 35 U.S.C. § 103(a) as obvious over Burton and Karedla. Pet. 20–35.

1. Summary of Burton

Burton is a U.S. patent titled “Method, System, and Program for Demoting Data From Cache Based on Least Recently Accessed and Least Frequently Accessed Data.” Ex. 1005, [54]. Burton issued May 18, 2004 from an application filed June 9, 2000, and is prior art to the ’745 under 35 U.S.C. § 102(e). *Compare id.* [22], [45], *with* Ex. 1001, [22], [45].

Burton describes as background a least recently used (LRU) algorithm for managing data in a cache. Ex. 1005, 1:39–49. Burton explains that when a track is added to the cache, a pointer to the track is placed at the top of the LRU linked list. *Id.* at 1:39–42. Burton explains that when the cache manager determines that data must be demoted or removed from cache to make room for subsequent data accesses, the cache manager will demote tracks having pointers at the bottom of the list, representing those tracks that were accessed the longest time ago relative to other tracks in the cache. *Id.* at 1:44–49. Burton states that “LRU algorithms are not optimal for randomly accessed data” because a demoted track in a cache, even though it is the least recently accessed, may be more likely to be subsequently accessed than those less frequently accessed. *Id.* at 1:50–58.

Burton discloses a caching method that “is an improvement over prior art techniques that do not take into account the frequency of access when demoting cache entries.” *Id.* at 2:31–34. According to Burton, “an entry that is more frequently accessed relative to another entry is more likely to be accessed again in the future.” *Id.* at 2:26–29. Burton discloses “select[ing]

for demotion those entries less likely to be accessed in the future with respect to other entries.” *Id.* at 2:29–31.

Figure 1 of Burton illustrates Burton’s cache architecture and is shown below:

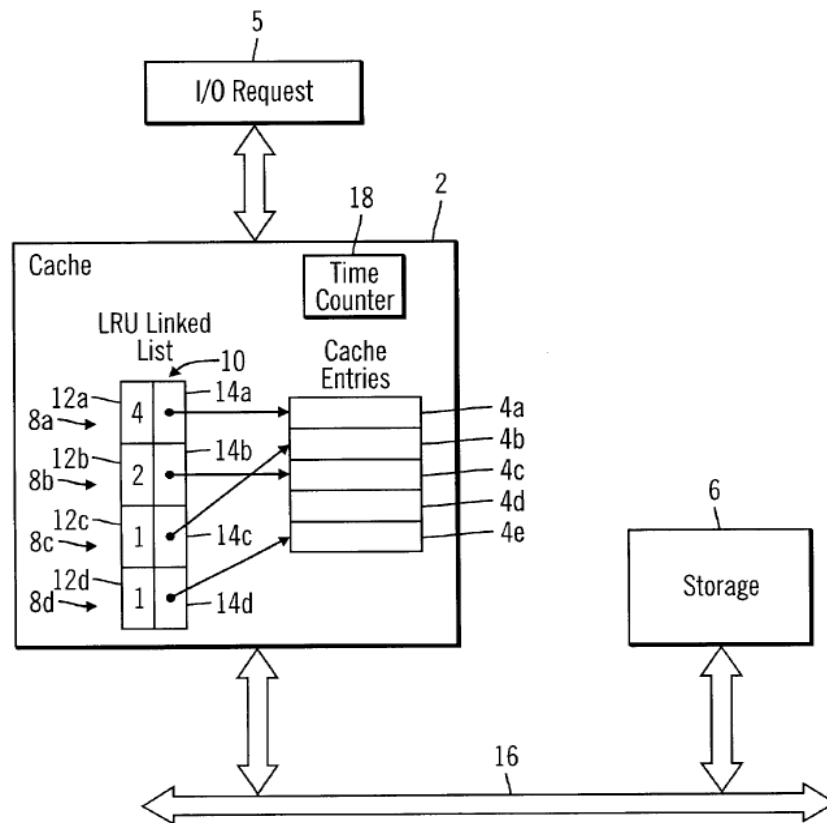


FIG. 1

Id. at 2:55–56. As illustrated in Figure 1 above, cache 2 includes cache entries 4a, 4b, 4c, 4d, into which tracks or pages of data from storage 6 may be placed. *Id.* at 2:55–58. Burton discloses that when a track or page of data is staged into a cache entry, new entry 8a is added to the top of LRU linked list 10. *Id.* at 2:58–60. According to Burton, for each cache entry 4a, 4b, 4c, 4d, LRU linked list 10 includes one entry 8a, 8b, 8c, 8d, respectively. *Id.* at 2:60–61. Burton discloses that each entry in LRU linked list 10 includes LRU rank value 12a, b, c, d and pointer 14 a, b, c, d to one cache entry 4a, b,

c, d, respectively. *Id.* at 2:62–66. Burton discloses that when an input/output request is serviced from the cache (i.e., a cache hit), then the entry in the LRU linked list including the pointer “is moved to the top of the LRU linked list.” *Id.* at 2:66–3:3.

Burton discloses that the LRU rank “provides a value for each entry in [the cache] that indicates both how frequently the entry is accessed and the time of last access.” *Id.* at 3:11–13. Burton states that “[t]he purpose of this weighting is to allow the process [of] demoting entries . . . from cache . . . to take into account the frequency with which an entry was accessed and remove those less frequently accessed entries.” *Id.* at 3:13–17 (reference numerals omitted). Burton describes its ranking methodology, which uses time counter 18 (shown in Figure 1 above), as follows:

The . . . LRU rank is weighted for previous accesses. Because the time counter 18 is incremented for every I/O access, the value added to the LRU rank for subsequently accessed cache entries increases substantially in a short period of time. Thus, LRU entries recently accessed or added to cache will have a substantially higher LRU rank than the LRU rank of entries that have not been accessed recently, even those entries accessed numerous times. However, to the extent entries have not been accessed for the same amount of time, the entry that was accessed more frequently will have a higher LRU rank because its LRU rank will be weighted with previous accesses. Thus, to the extent entries were last accessed at about the same time, which in terms of cache operations is within fractions of a second of each other, the more frequently accessed entry will have a greater weighting.

Id. at 3:54–4:2. Burton explains that the more frequently accessed entry is given a greater weighting because “a cache entry more frequently accessed in the past is likelier to be accessed in the future over less frequently

accessed entries that were last accessed at about the same time.” *Id.* at 4:2–5.

Burton also discloses that when the number of cached entries reaches an upper threshold, the cache determines from the last 1024 entries in the LRU linked list thirty-two entries that have the lowest LRU rank. *Id.* at 4:7–11. According to Burton, the cache then demotes those thirty-two entries from the cache. *Id.* at 4:11–13. Burton states that “those entries having the lowest LRU rank marked for demotion are both the least recently accessed and[,] among those entries least recently accessed, are less frequently accessed.” *Id.* at 4:19–22. Thus, according to Burton, “the algorithm . . . ensures that more frequently accessed cache entries . . . remain in cache longer.” *Id.* at 4:22–24.

2. Summary of Karedla

Karedla is an article from the March 1994 issue of *Computer* titled “Caching Strategies to Improve Disk System Performance.” Ex. 1006, 1, 6.⁷ The cover page and table of contents of Karedla indicated a date of March 1994, and the second page of Karedla has a stamp of the Copyright Office of the Library of Congress dated March 22, 1994. *Id.* at 1, 2, 4.

Karedla characterizes its disclosure as follows:

In this article, we examine the use of caching as a means to increase system response time and improve the data throughput of the disk subsystem. After explaining some basic caching issues . . . , we examine some popular caching strategies and cache replacement algorithms, as well as the advantages and disadvantages of caching at different levels of the computer

⁷ Although Petitioner cites to the original page numbers of Karedla rather than the exhibit page numbers added by Petitioner, for ease of reference to the entire exhibit, we cite to the exhibit page numbers.

system hierarchy. Finally, we investigate the performance of three cache replacement algorithms: random replacement (RR), least recently used (LRU), and a frequency-based variation of LRU known as segmented LRU (SLRU).

Id. at 6.

Karedla discloses that one of the main parameters in I/O cache design is the line replacement algorithm. *Id.* at 7. Karedla explains that vendors attempt to design efficient replacement algorithms that can offer higher cache performance for relatively small cache sizes. *Id.* Karedla describes “some of the more popular cache replacement algorithms,” including least recently used (LRU), least frequently used (LFU), and variations on LRU. *Id.* at 8. Karedla describes the LRU algorithm as one that “evicts the cache line used least in the recent past on the assumption that it won’t be used in the near future.” *Id.* Karedla also states “LRU uses only the time since last access and does not take into account the access frequency of lines when making replacement decisions.” *Id.* Karedla describes the LFU algorithm as one that “maintains a frequency-of-access count for all its lines and replaces the line that has been used least frequently.” *Id.* Karedla discloses that a disadvantage of the LFU algorithm is that “lines with large frequency counts that will not be accessed again (such as a recently active but currently cold file) tend to remain entrenched in the cache, preventing newer additions to the cache from gathering sufficient reference counts to stay in the cache.” *Id.*

Karedla describes variations on the LRU algorithm, stating that a growing trend is toward algorithms that “generally are combinations of LRU, LFU, and read-ahead strategies, with varying thresholds on the amount of data that is prefetched.” *Id.* Karedla discloses that “[a] read-

ahead strategy known as prefetching exploits the principle of spatial locality in an attempt to minimize latency in data access by anticipating future requests for data and bringing it into the cache.” *Id.*

One particular algorithm described in Karedla is “[a] frequency-based LRU replacement algorithm,” which Karedla states “improves on LRU by partitioning the LRU stack into three sections whose sizes are tunable.” *Id.* Karedla further discloses: “On being referenced, a line is placed in the topmost section of the stack. Unlike LFU, the reference count of lines repeatedly referenced in the top section is not incremented.” *Id.*

3. Analysis

a. Claim 4

Petitioner contends the combination of Burton and Karedla renders claim 4 obvious. Pet. 25–32. In the Final Written Decision of IPR2016-01643, the Board determined that the combination of Burton and Karedla renders claim 4 obvious. IPR2016-01643, Paper 51, 37–44. For the reasons given below, we reach the same determination here.⁸

“reading an extended segment of data”

Claim 4 is directed to “[a] caching method for enhancing system performance” and recites “reading an extended segment of data in response to a request from an operating system.” Petitioner argues that Burton describes “[a] caching method for enhancing system performance.” Pet. 25

⁸ Patent Owner has stated previously that “the Board’s consideration of the Burton/Karedla obviousness challenge in IPR2016-01643 will resolve the question of patentability of claim 4.” PO Prelim. Resp. 1–2. But, in its Supplemental Response, Patent Owner presents arguments for patentability of claim 4 over Burton and Karedla, which we address here, consistent with our Decision on this issue in IPR2016-01643.

(citing Ex. 1005, 1:8–11, 1:22–25, 4:30–34; Ex. 1002 ¶ 89). We agree because, as its title and field indicate, Burton is directed to a method for demoting data from cache. Ex. 1005, 1:8–11 (“The present invention relates to a method . . . for managing data in cache”); *see also id.* at 4:30–34 (disclosing that “the cache management scheme of the preferred embodiments has improved . . . over systems that employ the prior art”).

Petitioner argues that Karedla discloses the claimed “reading an extended segment of data in response to a request from an operating system” by describing a “read-ahead strategy known as prefetching” used in “*file system caches that prefetch the rest of a file that is being read*, and query optimizers in database applications.” Pet. 2526 (quoting Ex. 1006, 8). In particular, Petitioner contends that Karedla discloses caching strategies, including reading ahead, that exploit the principle that if a block of data is accessed, then nearby blocks of data will also soon be accessed. *Id.* at 26 (citing Ex. 1006, 7–8). Petitioner also contends Karedla discloses that prefetching exploits the principle of spatial locality by anticipating future requests and bringing them into the cache. *Id.* (citing Ex. 1006, 8; Ex. 1002 ¶ 91).

We find that Karedla discloses “reading an extended segment of data in response to a request from an operating system.” For example, Karedla discloses: “A read-ahead strategy known as prefetching exploits the principle of spatial locality in an attempt to minimize latency in data access by anticipating future requests for data and bringing it into the cache. Most disk drives implement prefetch in an onboard cache.” Ex. 1006, 8.

“storing copies of files associated with the extended segment”

Claim 4 further recites “storing copies of the files associated with the extended segment in a cache.” Petitioner argues that Karedla discloses this limitation by describing caches that prefetch the rest of a file that is being read. *Id.* at 27 (citing Ex. 1006, 8; Ex. 1002 ¶ 93). In describing “[b]asic caching terminology,” Karedla discloses that “[a] *cache buffer* is faster memory used to enhance the performance of a slower memory (a disk drive, for example), known as the backing store. By keeping copies of backing store data, caches can service some requests at faster memory speeds.” Ex. 1006, 7. Karedla also discloses “anticipating future requests for data and bringing it into the cache.” *Id.* at 8. Therefore, we determine Karedla stores files associated with the extended segment in the cache. Accordingly, we agree with Petitioner, and we find Karedla discloses “storing copies of the files associated with the extended segment in a cache.”

“assigning frequency factors to each of the files stored in the cache”

Petitioner argues that Burton discloses “assigning frequency factors to each of the files stored in the cache, the frequency factors indicating how often each of the corresponding files are requested by the operating system” as recited in claim 4 in describing a least recently used (LRU) linked list that includes an LRU rank value for each entry in the cache. Pet. 27–29 (citing Ex. 1005, Fig. 1, 2:3–5, 2:16–17, 2:60–63, 5:27–30). Burton discloses that for “each entry in cache, a variable indicates both a time when the cache entry was last accessed and a frequency of accesses to the cache entry,” and that “the variable for a cache entry is increased whenever the cache entry is accessed.” Ex. 1005, 2:3–5, 5:27–30, *cited in* Pet. 27.

Petitioner contends Burton further discloses weighting each LRU rank value for allowing the process to demote entries from cache “to take into account the frequency with which an entry was accessed and remove those less frequently accessed entries.” Pet. 28 (citing Ex. 1005, Fig. 1, 3:11–17). According to Petitioner, the weighting of Burton is similar to an embodiment of the ’745 patent that excludes the MRU from elimination. Pet. 29 (citing Ex. 1001, 7:59–61). Petitioner contends that Burton’s disclosure of incrementing the time counter for every access increases the LRU rank value for subsequently accessed cache entries in a short period of time. *Id.* (citing Ex. 1005, 3:55–58). Petitioner further contends that, as a result of such weighting, the MRU value can be distinguished from even frequently accessed entries, because recently accessed entries will have a higher LRU rank than the LRU rank of entries that have not been accessed recently. *Id.* (citing Ex. 1005, 3:58–61; Ex. 1002 ¶ 97).

Patent Owner asserts that Burton does not disclose “assigning frequency factors to each of the files,” as claim 4 requires. PO Supp. Resp. 13–25. Patent Owner further asserts Burton’s LRU rank value indicates a file’s recency of use, not frequency of use. *Id.* at 13. According to Patent Owner, Burton’s disclosure shows that recency of file requests dominates frequency of file requests in calculating the LRU rank value. *Id.* at 13–14 (citing Ex. 2021 ¶¶ 140–142). Patent Owner contends that the LRU rank of Burton shows, with a fair degree of certainty, how recently each of the files has been requested, not how often each of the files is requested. *Id.* at 14. Patent Owner explains its position using an example application of the algorithm illustrated in Burton’s Figure 2. *Id.* at 15–21 (citing Ex. 2021 ¶¶ 144–157). Patent Owner argues Burton discloses that the LRU rank for

the most recently accessed cache entry is incremented, per block 110 of Figure 2, “by the modulo (*i.e.*, the remainder) of the time counter divided by 512.” *Id.* at 16. In Patent Owner’s example, track A has the lowest LRU rank even though track A has been accessed more than tracks B through F. *Id.* at 17–21. According to Patent Owner, the LRU rank does not provide a reliable estimate of how often a file is requested by the operating system. *Id.* at 24 (citing Ex. 2021 ¶ 160).

Petitioner contends Burton discloses that the LRU rank values are a hybrid metric that indicate both how frequently an entry is accessed and the time of last access. Pet. Supp. Reply 15–16 (citing Ex. 1005 4:19–22 (“entries having the lowest LRU rank [are] marked for demotion [and] are *both* the least recently accessed *and among those entries least recently accessed recently, are less frequently accessed*”)); Ex. 1028 ¶¶ 101–102). Petitioner cites to Patent Owner’s Supplemental Response in persuasively contending that Patent Owner concedes that Burton’s LRU rank takes frequency into account such that, between two files that were last accessed at about the same time, the more frequently accessed entry will have a greater weighting. *Id.* at 17 (citing Pet. Supp. Resp. 14–16; Ex. 1028 ¶ 103).

The specification of the ’745 patent describes frequency factors that are based in part on frequency of use, and based in part on recency of use. See Ex. 1001, 6:2–4 (“the frequency factor is decreased with the lack of use of the corresponding file”), 6:56–7:51, 8:67–9:5, 12:52–53 (“decrementing a frequency factor after a time period of non-use of a corresponding file”). Dr. Chong provides testimony showing that a frequency factor based in part on frequency of use and based in part on recency of use may not show, with a fair degree of certainty, how often each of the corresponding files were

accessed. *See* Ex. 1026, 53:1–21 (“Q. And it [the frequency factor] can go -
- start at a hundred and decrement all the way down to two. That’s fair,
right? A. Potentially.”)).

We determine Burton’s algorithm of Figure 2 is weighted for previous accesses. Ex. 1005, 3:54; Ex. 1002 ¶ 96. The purpose of this weighting is to allow the process to take into account the frequency with which an entry was accessed and to remove those entries accessed less frequently. Ex. 1005, 3:13–17; Ex. 1002 ¶ 96. Burton’s algorithm of Figure 2 increments the LRU rank for the most recently accessed cache entry, such that the incremented LRU rank is (a) weighted for previous accesses, and (b) will have a substantially higher LRU rank than the LRU rank of entries that have not been accessed recently, even those accessed numerous times. *See* Ex. 1005, 3:55–4:2; Ex. 1002 ¶ 96.

We find the LRU rank of Burton is based in part on frequency of use, and in part on recency of use. *See* Ex. 1005, 3:11–13 (“The LRU rank . . . provides a value for each entry . . . in cache 2 that indicates both how frequently the entry is accessed and the time of last access”). We determine here, as we did in IPR2016-01643, that under the broadest reasonable construction, Burton’s LRU rank teaches “frequency factors indicating how often each of the corresponding files are requested by the operating system,” as recited in claim 4. This provides an indicator based on frequency of access and recency of access.

Petitioner also contends that Burton teaches this limitation even under Patent Owner’s narrower construction requiring that the “frequency factor” indicate frequency “with a fair degree of certainty.” Pet. Supp. Reply 17. According to Petitioner, Dr. Chong applied Burton’s algorithm during

initialization, which is incorrect, because eviction is not needed until many cycles after initialization, when the cache is full. *Id.* at 19–20 (citing Ex. 1027, 116:15–18; Ex. 1028 ¶ 106). Petitioner, providing an example using the same data access patterns used by Dr. Chong, persuasively shows that Burton’s algorithm, performed when the cache is full and eviction is needed, discloses the claimed frequency factor under Patent Owner’s narrower construction, to the same degree as the ’745 patent. *Id.* at 18–20 (citing Ex. 1028 ¶¶ 107–118). In Petitioner’s example, with a counter value of 100, entry A is accessed three times and has the highest LRU rank, entries B and C are each accessed twice and are both ranked below A and above D, E, and F, which are each accessed once. *Id.* at 18–19 (citing Ex. 1028 ¶¶ 107–108). Dr. Kubiawicz provides the following table showing the LRU rank for each entry after this series of accesses:

Entry	LRU Rank	Access Count
A	<u>3</u> 03	3
B	<u>2</u> 07	2
C	<u>2</u> 14	2
D	<u>1</u> 06	1
E	<u>1</u> 07	1
F	<u>1</u> 08	1

Ex. 1028 ¶ 107, *cited in* Pet. Supp. Reply 18. The table reproduced above shows cache entries with LRU ranks corresponding to the number of accesses. For example, entry A, which was accessed 3 times, has an LRU

rank of 303, while entries that were accessed twice have LRU ranks in the 200s, and entries that were accessed once have LRU ranks in the 100s.

Dr. Kubiadowicz testifies that Burton's algorithm, applied when the cache is full, results in the same behavior as a self-tuning embodiment of the '745 patent that decreases existing rank values based on lack of use.

Ex. 1028 ¶¶ 111–112, 117–118. Thus, in addition to finding that Burton teaches this limitation under its broadest reasonable construction, we rely on the testimony of Dr. Kubiadowicz and determine that the LRU rank of Burton teaches this limitation even under Patent Owner's narrower construction requiring that the "frequency factor" indicate frequency "with a fair degree of certainty."

"scanning the frequency factors"

Petitioner argues that Burton discloses "scanning the frequency factors, the scanning being performed in response to a target capacity of the cache being attained," in describing determining when a number of cache entries has reached a threshold, and identifying entries that have the lowest LRU rank. Pet. 29–31 (citing Ex. 1005, Fig. 3, 2:6–11, 4:6–13; Ex. 1002 ¶¶ 98–100). Dr. Kubiadowicz testifies that in response to reaching the threshold, the cache of Burton determines, from the last 1024 entries in the LRU linked list, thirty-two entries that have the lowest LRU rank and then demotes those thirty-two entries from the cache. Ex. 1002 ¶ 100 (citing Ex. 1005, 4:8–13, 4:19–22). We agree with Petitioner, and based on the disclosure of Burton (Ex. 1005, 4:7–13, 19–22) and the testimony of Dr. Kubiadowicz, we find that Burton teaches the recited limitation of "scanning the frequency factors, the scanning being performed in response to a target capacity of the cache being attained."

“identifying a least frequently and least recently used file; and eliminating the least frequently and least recently used file to liberate capacity of the cache”

Petitioner, relying on testimony of Dr. Kubiadowicz, argues that Burton discloses “identifying a least frequently and least recently used file” in describing identifying a plurality of cache entries having a relatively low value with respect to the value for other cache entries, where the relatively low value indicates that the cache entry is one of a least recently access entry and/or least frequently accessed entry. Pet. 31 (citing Ex. 1005, 2:9–15, 4:9–11; Ex. 1002 ¶ 101). Petitioner, relying on testimony of Dr. Kubiadowicz, argues that Burton discloses “eliminating the least frequently and least recently used file to liberate capacity of the cache” in describing that the entries having the relatively low value are demoted from cache to make room for subsequently accessed data. Pet. 31 (citing Ex. 1005, 2:11–15, 2:23; Ex. 1002 ¶ 102).

We agree with Petitioner, and based on the disclosure of Burton (Ex. 1005, 2:9–15, 2:23, 4:9–13) and the testimony of Dr. Kubiadowicz, we find that Burton teaches “identifying a least frequently and least recently used file; and eliminating the least frequently and least recently used file to liberate capacity of the cache.”

b. Rationale to Combine Asserted Art

Petitioner also asserts that a person of ordinary skill in the art would have been motivated to implement Karedla’s method of prefetching in Burton’s cache architecture, to further decrease latency and increase performance in a computer architecture system that implements the caching

architecture of Burton. Pet. 33–35 (citing Ex. 1002 ¶¶ 110–113; Ex. 1005, 1:22–27, 2:25–27; Ex. 1006, 8).

We find Burton teaches determining which entries to demote from cache to make room for subsequently accessed data. Ex. 1005, 2:23–27. We find Karedla teaches caching strategies to improve system performance, including a read-ahead strategy known as prefetching. Ex. 1006, 8. We find Dr. Kubiawicz persuasively explains the benefits and predictable results of combining the teachings of Burton and Karedla, such as decreasing latency and increasing system performance. Ex. 1002 ¶¶ 110–113. Thus, we find the Petition and supporting evidence articulates a reason with a rational underpinning for combining the teachings of Burton and Karedla as claimed—to further decrease latency and increase system performance.

Accordingly, we determine that Petitioner shows, by a preponderance of the evidence, that claim 4 is unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of Burton and Karedla.⁹

c. Claim 5

Petitioner contends that the combination of Burton and Karedla teaches “files are read from a storage medium containing one of a directory, a file allocation table and a data file” as recited in claim 5. Pet 32. Petitioner contends Karedla discloses file system caches that prefetch the rest of a file that is being read. *Id.* (citing Ex. 1006, 8; Ex. 1002 ¶ 105); see Ex. 1006, 8 (describing “file system caches that prefetch the rest of a file that is being read”).

⁹ Patent Owner does not present any objective evidence of nonobviousness as to any of the challenged claims.

We agree with Petitioner, and based on the disclosure of Burton and the testimony of Dr. Kubiawicz, we determine that the combination of Burton and Karedla teaches this limitation.

d. Claim 6

Claim 6 recites “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file.” Petitioner contends that Burton discloses an LRU linked list, where the top of the list holds the MRU cache entry, and the bottom of the list holds the LRU entry. Pet. 32–33 (citing Ex. 1005, 1:47–49, 2:58–60; Ex. 1002 ¶¶ 107–108). Petitioner contends that Burton discloses scanning the last 1024 entries of the LRU linked list to determine thirty-two entries that have the lowest LRU rank. *Id.* at 33 (citing Ex. 1005, 4:9–11). Petitioner, relying on testimony of Dr. Kubiawicz, contends that a person of ordinary skill would have understood that the scanning occurs from the LRU entry at the bottom of the linked list to the top of the linked list, where the MRU entry is. Pet 33 (citing Ex. 1002 ¶ 109).

Patent Owner presents arguments that rely on its claim construction that “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file” as recited in claim 6 requires scanning all files from the LRU to the MRU, inclusive. PO Supp. Resp. 4–6. As we discussed above in our construction of this phrase, we do not agree with Patent Owner’s construction. Rather, the “scanning” limitation, properly construed, encompasses scanning in a direction that proceeds from the LRU file towards the MRU file.

We find Burton discloses adding a new cache entry to the top of an LRU linked list. Ex. 1005, 2:58–60. We find Burton discloses determining,

from the last 1024 entries in the LRU linked list, thirty-two entries that have the lowest LRU rank. Ex. 1005, 4:9–11. We rely on testimony of Dr. Kubiadowicz to determine that a person of ordinary skill in the art would have understood that, in order to find the thirty-two entries, the LRU linked list is scanned in a direction that proceeds from the LRU entry of the linked list to the MRU entry of the linked list. Ex. 1002 ¶¶ 107–109.

We agree with Petitioner, and based on the disclosure of Burton and the testimony of Dr. Kubiadowicz, we determine that the combination of Burton and Karedla teaches this limitation.

Petitioner also contends that even under Patent Owner’s narrower construction, when a cache has 1024 entries or less, Burton’s algorithm would scan all files from the LRU file to the MRU file, inclusive. Pet. Supp. Reply 21–22 (citing Ex. 1026, 145:17–146:8; Ex. 1027, 26:6–28:19, 27:16–19; Ex. 1028 ¶¶ 120–122; Ex. 1001, 4:9–11). Dr. Kubiadowicz testifies that in “caches of this scale (under 1024 blocks), there is no dispute that Burton would scan ‘all files from the LRU file to the MRU file, inclusive.’” Ex. 1028 ¶ 121.

We rely on Dr. Kubiadowicz’s testimony and determine that even under Patent Owner’s construction, Burton teaches “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file” as claimed for at least cache sizes under 1024 blocks. *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016) (“It does not matter that the use of alphabetical order for locations would not always result in farther-over-nearer ordering. It is enough that the combination would sometimes perform all the method steps, including farther-over-nearer ordering.”).

e. Summary

For the above reasons, we determine Petitioner shows, by a preponderance of the evidence, that claims 4–6 are unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of Burton and Karedla.

E. Asserted Obviousness over Lee, Dharap, and Karedla

Petitioner contends that claims 4 and 5 are unpatentable under 35 U.S.C. § 103(a) as obvious over Lee, Dharap, and Karedla. Pet. 43–61. Relying in part on the testimony of Dr. Kubiawicz, Petitioner explains how the references teach or suggest the claim limitations and provides reasoning for combining the teachings of the references. *Id.* (citing Ex. 1002 ¶¶ 137–70).

1. Lee

Lee is directed to the implementation and evaluation of a caching mechanism utilizing a “Least Recently/Frequently Used” method for eliminating files from the cache (“the LRFU replacement policy”). Ex. 1003, Abstract. As background, Lee teaches that the “speed gap between processors and disks is becoming wider” and that “[o]ne solution to overcoming this speed gap is to use a caching technique which keeps disk blocks that are likely to be accessed in the near future in DRAM memory.” *Id.* at 106. But, Lee explains, “[s]ince the buffer cache size is necessarily limited, an effective scheme is needed for the block replacement policy that decides which block should be replaced.” *Id.*

Lee discloses that the “[p]revious research on block replacement policies [for caches] can be divided into two groups.” *Id.* The “Least Frequently Used (LFU)” policy is an example of “policies [that] base their

replacement decision on the frequency of references.” *Id.* And the “Least Recently Used (LRU)” policy is an example of “policies [that] base their replacement decision” “on the recency of references.” *Id.* Lee teaches that, “[r]ecently, a novel block replacement policy called the LRFU (Least Recently/Frequently Used) policy was proposed that subsumes both the LRU and LFU policies, and provides a spectrum of block replacement policies between them.” *Id.*

Lee explains that the LRFU replacement policy “associates a value” “called the CRF (Combined Recency and Frequency) value” “with each block” that “quantifies the likelihood that the block will be referenced in the near future.” *Id.* at 107. “Each reference to a block in the past adds its contribution to this [CRF] value and its contribution is determined by [a] *weighing function . . .*” *Id.* The weighing function includes “a control parameter that determines a trade-off between recency and frequency.” *Id.* In other words, the control parameter controls the relative contributions of recency and frequency to the CRF values. *See id.* “The LRFU policy replaces the block whose CRF value is the smallest in the cache.” *Id.*

2. Dharap

Dharap teaches “information processing systems that utilize cache memory to minimize latency.” Ex. 1004 ¶ 2. Dharap provides as background a description of cache systems. “Traditionally,” Dharap explains, “cache memory is filled with copies of information resources that a user receives from a remote source, ‘remote’ being defined as being further removed from the user than the cache memory, e.g., local main memory or a server in a client-server architecture.” *Id.* ¶ 4. Dharap describes the timesaving benefit of this system: “If the user subsequently requests the

same resource, the resource's copy is provided from the cache memory, rather than from the original remote source, thereby saving the time required to receive the resource from the remote source for a second time.” *Id.*

Dharap explains, that “[w]hen the cache memory becomes full,” however, the cache controller must “make room for copies of new resources that the user accesses.” *Id.* Dharap then explains that “[a] variety of criteria, commonly termed caching policies, are available to determine which resource copy to remove from cache memory” to make room for new information. *Id.* “Such caching policies can be based on: the duration since the last access, the number of times accessed since originally received, the amount of memory allocated to the resource, the difficulty of retrieving the resource from the remote site, etc.” *Id.*

Dharap teaches that “the cache controller in a conventional system also removes copies of resources from the cache memory when it is predicted or determined that the source information has changed, because the copy of the resource in the cache memory is outdated, or ‘stale.’” *Id.*

¶ 5. Dharap points to Internet content as one example of information that may become outdated. *Id.*

Dharap discloses “a cache system that caches copies of resources based on the semantic type of the resource.” *Id.* ¶ 9. Dharap explains:

The expression “semantic type” as used within this context refers to the different connotative meanings that the information contents of resources can have, as perceived by the user. For example, some information content may be perceived as highly volatile (e.g., being of short term relevance such as web sites dedicated to the results of sports matches, to specific stock market news or currency exchange rates), other information content may be perceived as rather static (e.g., being of long-term relevance such as glossaries on the Internet). Semantic

types that can be expected to contain dynamic information, such as news Web sites and weather Web sites, need a caching policy wherein the copy in the cache memory is selected for replacement based upon the duration of time that the copy has been in the cache memory. Conversely, semantic types that can be expected to relate to static resources, such as encyclopedic information, glossaries, etc., need a more conservative caching policy, such as least-recently-used (LRU) or least-frequently used (LFU), that are substantially independent of the time duration that the copy remains in the cache memory.

Id. Dharap also teaches that “a combination of caching policies” may be employed for certain types of resources. *Id.*

3. Analysis

a. Claim 4

“reading an extended segment of data in response to a request from an operating system”

Petitioner contends that Karedla discloses this limitation. Pet. 47 (citing Pet. 25–26; Ex. 1002 ¶¶ 147–148). As we discussed above in our analysis of the combination of Burton and Karedla, we find that Karedla discloses “reading an extended segment of data in response to a request from an operating system.” For example, Karedla discloses: “A read-ahead strategy known as prefetching exploits the principle of spatial locality in an attempt to minimize latency in data access by anticipating future requests for data and bringing it into the cache. Most disk drives implement prefetch in an onboard cache.” Ex. 1006, 8.

“storing copies of files associated with the extended segment in a cache”

Petitioner, relying on testimony of Dr. Kubiawicz, contends that Lee discloses this limitation in describing a buffer cache that keeps disk blocks

that are likely to be accessed in the near future. Pet 47–48 (citing Ex. 1003, 106; Ex. 1002 ¶ 149). We rely on Dr. Kubiawicz’s testimony and find that Lee discloses this limitation.

“assigning frequency factors to each of the files stored in the cache, the frequency factors indicating how often each of the corresponding files are requested by the operating system”

Petitioner, relying on testimony of Dr. Kubiawicz, contends Lee discloses this limitation in describing assigning a value to each file in cache that indicates how often the corresponding file is referenced, among other things. Pet. 48 (citing Ex. 1003, 106–107; Ex. 1002 ¶ 151). Lee discloses a combined recency and frequency (CRF) value to quantify the likelihood that the block will be referenced in the near future. Ex. 1003, 106. Each reference to a block in the past adds its contribution to this value. *Id.*

Patent Owner contends that the CRF value disclosures of Lee do not teach both the “frequency factors” limitation and the “eliminating the least recently used file” limitations. PO Resp. 37–56. Patent Owner contends that Lee discloses a control parameter, λ , to change the relative weights given to frequency of use and recency of use when calculating the CRF value. PO Resp. 40 (citing Ex. 1003, 107). Patent Owner contends that the least frequently/recently used (LFRU) algorithm can heavily weight the CRF value toward recency by choosing a λ value close to 1, and heavily weight the CRF value toward frequency by choosing a λ value close to 0. *Id.*

Patent Owner, relying on testimony of Dr. Chong, contends that choosing a λ value of 10^{-4} or smaller results in a CRF value that closely approximates the number of accesses of a block. PO Resp. 45 (citing Ex. 2004 ¶¶ 83–84).

Petitioner, relying on Dr. Chong's testimony cited by Patent Owner, persuasively argues that even under Patent Owner's construction, Lee discloses a frequency factor. Pet. Reply 16–17 (citing PO Resp. 45; Ex. 2004 ¶¶ 83–84). Petitioner highlights that Dr. Chong's example shows block A is accessed twice and has a CRF value of 1.99931, and block B is accessed three times and has a CRF value of 2.99792. *Id.*

We find Lee discloses a combined recency and frequency (CRF) value, where each reference to the block in the past adds its contribution to the CRF value. Ex. 1003, 106. We determine that “assigning frequency factors to each of the files stored in the cache, the frequency factors indicating how often each of the corresponding files are requested by the operating system” is taught by the CRF value of Lee, which is based on frequency of use.

“scanning the frequency factors, the scanning being performed in response to a target capacity of the cache being attained”

Petitioner contends the teachings of Lee and Dharap teach this limitation. Pet. 49. Petitioner, relying on testimony of Dr. Kubiawicz, contends that Lee teaches ordering blocks according to their CRF values, and replacing blocks because of the limited buffer cache size. Pet. 49 (citing Ex. 1003, 106–107; Ex. 1002 ¶ 153). Petitioner, relying on testimony of Dr. Kubiawicz, contends that although Lee does not explicitly teach attaining a target capacity, Dharap teaches removing cache items in response to a target capacity of the cache, such as when the cache is full. Pet. 49–50 (citing Ex. 1004 ¶ 4; Ex. 1002 ¶ 154). We rely on Dr. Kubiawicz's testimony and determine that the combined teachings of Lee and Dharap teach this limitation.

“identifying a least frequently and least recently used file; and eliminating the least frequently and least recently used file to liberate capacity of the cache”

Petitioner contends Lee teaches this limitation in disclosing identifying a cache item with the smallest combined CRF value and eliminating it from the cache. Pet. 50–51 (citing Ex. 1003, 106–108). Petitioner, relying on testimony of Dr. Kubiawicz, contends that the item with the smallest CRF value is the least frequently and least recently used item. *Id.* at 51 (citing Ex. 1003, 107; Ex. 1002 ¶ 155).

Patent Owner contends that even if Lee’s CRF value is a frequency factor, eliminating the file with the lowest CRF does not necessarily eliminate a “least recently used file.” PO Resp. 38. In particular, Patent Owner cites Dr. Chong’s example of Lee’s LRFU algorithm eliminating a most recently used block when λ is 10^{-4} or smaller, which, according to Patent Owner, cannot be “a least recently used file.” *Id.* at 45–48 (citing Ex. 2004 ¶¶ 83–88). Patent Owner contends that even at higher values of λ , Lee’s CRF value often identifies the most recently used file for elimination. *Id.* at 50–54 (citing Ex. 2004 ¶¶ 92–99).

Petitioner contends that Patent Owner’s contention is premised on an erroneous construction of “a least frequently and least recently used file.” Pet. Reply 13. As we discussed in our claim construction above, the proper construction of this term encompasses an embodiment in the specification of the ’745 patent in which the MRU can be identified and eliminated as a least frequently and least recently used file.

We rely on Dr. Chong’s testimony and the embodiment disclosed in 6:19–21 of the ’745 patent, and determine that “identifying a least frequently

and least recently used file; and eliminating the least frequently and least recently used file to liberate capacity of the cache,” as claimed, is disclosed by Lee’s description of eliminating the file with the smallest CRF value.

Petitioner, relying on testimony of Dr. Kubiadowicz and Dr. Chong, further contends that even under Patent Owner’s narrower construction of never evicting the MRU file, the MRU file of Lee can never be evicted. Pet. Reply 18–20 (citing Ex. 1003, 108–109, Fig. 2(a); Ex. 1025 ¶¶ 45, 51–59; Ex. 1026, 138:13–139:16). Dr. Kubiadowicz testifies that Figure 2 of Lee shows three scenarios, (a), (b), and (c), for referencing a file. Ex. 1025 ¶ 54. Dr. Kubiadowicz testifies that, from the standpoint of whether Lee eliminates a most recently used file from the cache, scenario (a) is the only scenario that eliminates a file from the cache. *Id.* Dr. Kubiadowicz testifies that Figure 2(a) of Lee shows placing files with larger CRF values in a heap, and ordering files with smaller CRF values in a linked list, where the file with the smallest CRF value is at the end of the linked list. Ex. 1025 ¶ 53. Dr. Kubiadowicz testifies that Figure 2(a) shows that a new cache entry (the most recently used file) is placed in the heap, while the file at the end of the linked list is discarded. Ex. 1025 ¶¶ 54–55. According to Dr. Kubiadowicz, “because in Lee’s implementation the MRU file is in the heap and the file evicted from the cache is at the left end of the linked list, the cache eviction policy implementation disclosed in Lee *never evicts the MRU file from the cache.*” *Id.* ¶ 55. Rather, “the most rapid path to eviction for a MRU file happens when a file is accessed for the first time (making it the MRU) and placed at the root of the heap, and then gradually decay out of the cache moving through the linked list.” *Id.*

Lee discloses a modified buffer cache structure for an optimized implementation of the LRFU policy, where the cache structure has a linked list and heap combination. Ex. 1003, 108, Fig. 2. Dr. Chong testifies that for all λ values other than 0 or 1, the linked list and heap combination of Lee would be used. Ex. 1026, 140:1–15. Lee discloses that “[i]f the requested block is not found in the cache, the block with the smallest CRF value is replaced, and the root block at the heap is demoted to the linked list. Then, the requested block is inserted into the heap and the heap is restored.” Ex. 1003, 108, Fig. 2(a). Both Dr. Chong and Dr. Kubiawicz testify that Lee’s implementation places the MRU file in the heap and does not evict the MRU file. Ex. 1025 ¶¶ 55, 59; Ex. 1026, 138:13–139:16.

Relying on the testimony of Dr. Chong and Dr. Kubiawicz, and relying on Lee’s description of inserting the most recently used file into the heap of the cache, we determine that Lee does not evict the MRU file. We determine Lee would not evict the MRU file unless it gradually decayed out of the cache by moving to the end of the linked list, at which point the file would no longer be the MRU file. We determine that even under Patent Owner’s narrower construction of never evicting the MRU file, “identifying a least frequently and least recently used file; and eliminating the least frequently and least recently used file to liberate capacity of the cache” is disclosed by Lee’s description of eliminating the file with the smallest CRF value.

b. Rational to Combine

Petitioner, relying on testimony of Dr. Kubiawicz, contends that a person of ordinary skill in the art would have incorporated Dharap’s teaching of removing cache items when the cache is full when implementing

Lee's cache replacement policy for the benefit of minimizing cache latency and improving system performance as taught by Dharap. Pet. 52–55 (citing Ex. 1002 ¶¶ 161–166; Ex. 1004 ¶¶ 2, 4; Ex. 1003, 106–107).

Petitioner, relying on testimony of Dr. Kubiawicz, contends that a person of ordinary skill would have incorporated Karedla's teaching of prefetching into the cache system of Lee modified by Dharap for the benefit of further decreasing latency and increasing performance of the cache as taught by Karedla. Pet. 55–57 (citing Ex. 1002 ¶¶ 167–170; Ex. 1003, 107, 109–110; Ex. 1006, 8).

We find Dharap teaches using cache memory to minimize latency, including removing files from cache when the cache becomes full. Ex. 1004 ¶¶ 2, 4. We find Lee teaches a file replacement policy for cache that improves performance. Ex. 1003, 106–107. We find Karedla teaches caching strategies to improve system performance, including a read-ahead strategy known as prefetching. Ex. 1006, 8. We find Dr. Kubiawicz persuasively explains the benefits and predictable results of combining the teachings of Dharap, Lee, and Karedla, such as decreasing latency and increasing system performance. Ex. 1002 ¶¶ 161–170. Thus, we find the Petition and supporting evidence articulates a reason with a rational underpinning for combining the teachings of Dharap, Lee, and Karedla as claimed—to further decrease latency and increase system performance.

c. Claim 5

“the files are read from a storage medium containing one of a directory, a file allocation table and a data file”

Petitioner, relying on testimony of Dr. Kubiawicz, contends that Lee teaches this limitation in describing a cache in DRAM that stores files

Figure 1 above illustrates “how a combination of block aging, boundary condition and reference count techniques may be used.” *Id.* at 3:37–39. Robinson ’885 teaches that each cache directory entry **12** in cache directory **10** has a “reference count” associated with it. *Id.* at 4:24–26. According to Robinson ’885, when a new block is brought into the cache, the associated cache directory entry is placed into most recently used (MRU) position **14** and the reference count is initialized to one. *Id.* at 4:24–31. Robinson ’885 discloses that “the cache directory essentially works in LRU fashion, with a cache directory entry being put in the MRU position each time a block is referenced.” *Id.* at 4:34–37. According to Robinson ’885, however, “the block associated with the cache directory entry in the LRU position **16** will not necessarily be the one that is replaced when there is a miss.” *Id.* at 4:37–40.

As shown in Figure 1, the cache directory has age boundary 18. The section of the cache directory “on the MRU side of the boundary” is the “local” section; the section on the other side of the boundary is the “nonlocal” section. *Id.* at 4:46–47. Robinson ’885 discloses that “when there is a hit on the local section, the count remains the same; when there is a hit on the non-local section, the count is incremented.” *Id.* at 4:50–52. Robinson ’885 discloses ways of using the reference counts to select blocks to replace, including “select[ing] the least recently used block in the non-local section whose count is below a preselected threshold.” *Id.* at 4:55–57.

Figure 2 of Robinson ’885, shown below, illustrates an alternative embodiment in which the cache directory is divided into three sections using two boundaries:

FIG. 2

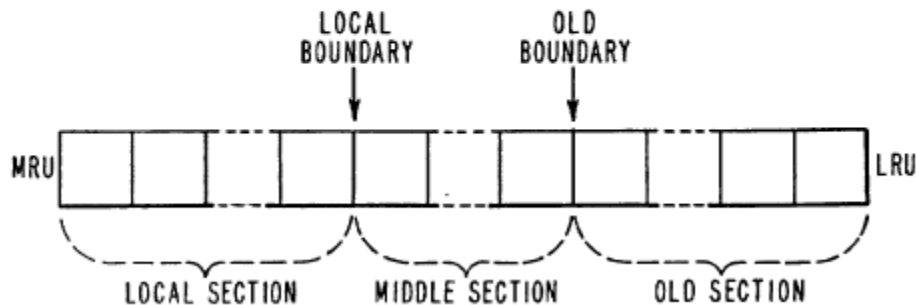


Figure 2 above shows a cache directory having a local section, a middle section, and an old section. *Id.* at 4:66–5:1. Robinson '885 discloses that “[b]locks with reference counts in the old section are the blocks from which replacement selections are made.” *Id.* at 5:1–3. Specifically, Robinson '885 discloses that “a block is selected to be replaced by finding the block (or blocks) with the smallest count in the old section and then replacing that block (or least recently used such block if there is more than one) if the count is below a predetermined threshold.” *Id.* at 5:4–8. 78. Robinson '885 further describes maintaining a separate LRU chain for blocks with a count of 1, another separate LRU chain for blocks with a count of 2, and so on. *Id.* at 5:10–14. Robinson '885 discloses finding a block to replace by “scanning the blocks (from LRU to MRU) in each such LRU chain (in ascending count order) until a block is found in the old section.” *Id.* at 5:15–18.

2. Analysis

Claim 6 depends from claim 4 and further recites “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file.” Ex. 1001, 13:8–11. Petitioner asserts that Robinson '885 teaches the claimed scanning by disclosing that “[f]inding a

block to replace . . . consists of scanning the blocks (from LRU to MRU) in each such LRU chain (in ascending count order) until a block is found in the old section.” Pet. 59 (quoting Ex. 1008, 5:15–18); *see also* Ex. 1002 ¶ 177 (same).

Patent Owner contends that Robinson scans blocks in only the old section of cache but does not scan any blocks in the middle section, nor the local section. PO Supp. Resp. 38–39 (citing Ex. 2021 ¶ 190). Patent Owner contends that Robinson ’885 does not scan all files from the LRU file to the MRU file, inclusive. *Id.* Patent Owner contends that the Board’s analysis in the original Decision to Institute was correct that Robinson ’885 does not disclose “the scanning from a LRU file to a MRU file” recited in claim 6. *Id.* at 39–40 (citing Paper 11, 26–27; Ex. 2021 ¶ 191).

Petitioner contends that Patent Owner’s argument is based on an improper construction of the “scanning” limitation of claim 6. Pet. Supp. Reply 23. Petitioner contends that Patent Owner does not dispute that Robinson ’885 discloses scanning in a direction from the LRU file to the MRU file. *Id.* According to Petitioner, claim 6, properly construed, is taught by Robinson ’885’s disclosure of scanning in a direction from LRU to MRU. *Id.*

As we discuss in our claim construction above, the broadest reasonable interpretation of “scanning from a frequency factor corresponding to a LRU file to a frequency factor corresponding to a MRU file” encompasses at least “scanning in a direction that proceeds from the LRU file towards the MRU file.” We determine that this limitation is taught by Robinson ’885’s disclosure of “scanning the blocks (from LRU to MRU)

in each such LRU chain (in ascending count order) until a block is found in the old section.” Ex. 1008, 5:15–18.

Petitioner, relying on Dr. Kubiawicz, contends that the combination of Lee, Dharap, Karedla, and Robinson, is the combination of known structures in a simple manner that yields the predictable results of a caching method for (i) reading extended data as taught by Karedla, (ii) replacing data based on identifying least recently and least frequently accessed data as taught by Lee and Dharap, and (iii) scanning the cache entries from the LRU entry to the MRU entry to identify candidate entries for replacement as taught by Robinson. Pet. 60–61 (citing Ex. 1003, 107; Ex. 1004 ¶ 2; Ex. 1006, 8; Ex. 1008, 1:19–23; Ex. 1002 ¶¶ 178–182). As discussed above, we find a person of ordinary skill in the art would have had reason to combine the teachings of Dharap, Lee, and Karedla. See Section II.E.3.b above. We find the Petition and supporting evidence show that combining Robinson ’885’s teachings of scanning in a particular direction with the teachings of Lee, Dharap, and Karedla is the “combination of familiar elements according to known methods [that] does no more than yield predictable results.” *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 416 (2007); see Ex. 1002 ¶¶ 178–182.

Accordingly, we determine Petitioner shows, by a preponderance of the evidence, that claim 6 is unpatentable under 35 U.S.C. § 103(a) as obvious over the combination of Dharap, Lee, Karedla, and Robinson.

F. Asserted Obviousness Over (a) Burton and Sweeney, (b) Lee, Dharap, and Sweeney, and (c) Lee, Dharap, Sweeney, and Robinson

Petitioner submits, and we agree, that these grounds are moot because they were raised only to address a potential claim construction for which

Patent Owner does not advocate. See Pet. Supp. Reply 1 n.1, 24. Therefore, we do not reach these grounds of unpatentability.

III. CHALLENGE TO PETITIONER’S REPLY ARGUMENTS

With Board authorization, Patent Owner filed a list of alleged improper reply arguments (Paper 30). Patent Owner lists several portions of Petitioner’s Reply and evidence allegedly beyond the scope of what can be considered appropriate for a reply. See Paper 30 (citing Reply).

We have considered Patent Owner’s listing, but we disagree that the cited portions of Petitioner’s Reply and reply evidence are beyond the scope of what is appropriate for a reply. For example, according to Patent Owner, Petitioner’s contention that the ’745 patent does consider the MRU file as a “least frequently and least recently used file” as claimed was improperly raised in the Reply. Paper 30, 1–2. We disagree. We determine that Petitioner properly made this contention to address Patent Owner’s argument that the ’745 patent does not consider the MRU file. See Reply 12; PO Resp. 30. As another example, Patent Owner contends that Petitioner improperly raised the argument that Lee eliminates the left-most block in a linked list. Paper 30, 3–5. We determine Petitioner properly made this contention to address Patent Owner’s argument that Lee evicts an MRU file fairly often. See Reply 18–20; PO Resp. 42, 45–52, 56. Patent Owner also contends that Petitioner improperly raised *Unwired Planet* in the Reply. Paper 30, 6–7. We determine Petitioner properly cited *Unwired Planet* to address Patent Owner’s argument that it is immaterial if Lee sometimes embodies the claimed method. See Reply 22–23; PO Resp. 54–56.

Replies are a vehicle for responding to arguments raised in a corresponding Patent Owner Response. Petitioner’s arguments and evidence are not beyond the proper scope of a reply because we find that they fairly respond to Patent Owner’s arguments raised in Patent Owner’s Response. *See Idemitsu Kosan Co., LTD. v. SFC Co. LTD*, 870 F.3d 1376, 1381 (Fed. Cir. 2017) (“This back-and-forth shows that what [Patent Owner] characterizes as an argument raised ‘too late’ is simply the by-product of one party necessarily getting the last word. If anything, [Patent Owner] is the party that first raised this issue, by arguing—at least implicitly—that [a reference] teaches away from non-energy-gap combinations. [Petitioner] simply countered, as it was entitled to do.”); *Anacor Pharms., Inc. v. Iancu*, 889 F.3d 1372, 1380 (Fed. Cir. 2018) (“In addition, the petitioner in an inter partes review proceeding may introduce new evidence after the petition stage if the evidence is a legitimate reply to evidence introduced by the patent owner”).

IV. CONCLUSION

On this record, Petitioner has shown by a preponderance of the evidence that claims 4–6 of the ’745 patent are unpatentable.

V. ORDER

Accordingly, it is:

ORDERED that

1. Claims 4–6 of the ’745 patent are unpatentable on the ground of obviousness over Burton and Karedla;
2. Claims 4 and 5 of the ’745 patent are unpatentable on the ground of obviousness over Lee, Dharap, and Karedla; and

3. Claim 6 of the '745 patent is unpatentable on the ground of obviousness over Lee, Dharap, Karedla, and Robinson; and

FURTHER ORDERED that because this Final Written Decision is final, a party to the proceeding seeking judicial review of the Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2017-00429
Patent 6,775,745 B1

PETITIONER:

Peter Dichiara
Theodoros Konstantakopoulos
WILMER HALE
peter.dichiara@wilmerhale.com
theodoros.konstantakopoulos@wilmerhale.com

Christopher Centurelli
Benjamin Weed
K&L GATES
christopher.centurelli@klgates.com
benjamin.weed.ptab@klgates.com

Thomas Brown
EMC CORPORATION
tom.brown@emc.com

PATENT OWNER:

Brenton Babcock
Ted M. Cannon
John R. King
KNOBBE, MARTENS, OLSON & BEAR, LLP
2brb@knobbe.com
2tmc@knobbe.com
2jrk@knobbe.com

Tim Seeley
James Hietala
INTELLECTUAL VENTURES
tims@intven.com
jhietala@intven.com