UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/597,174	01/14/2015	Frederic H. Tudor	058752-01-5213-US	1097

82474 7590 03/12/2020
Morgan, Lewis & Bockius LLP (PH)(SanDisk)
1701 Market Street
Philadelphia, PA 19103-2921

EXAMINER	
KHAN, MASUD K	

ART UNIT	PAPER NUMBER
2131	

NOTIFICATION DATE	DELIVERY MODE
03/12/2020	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

judith.troilo@morganlewis.com
phpatentcorrespondence@morganlewis.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte FREDERIC H. TUDOR, HARIHARA KADAYAM,
BRIAN W. O'KRAFKA, and JOHANN GEORGE

Appeal 2019-001307
Application 14/597,174
Technology Center 2100

Before JOHNNY A. KUMAR, SCOTT B. HOWARD, and MICHAEL T. CYGAN, *Administrative Patent Judges*.

KUMAR, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF CASE

Introduction

Pursuant to 35 U.S.C. § 134(a), Appellant[1] appeals the Final Rejection of claims 1–25. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

Exemplary Claim

Exemplary claim 1 under appeal reads as follows:

> 1. A method of managing a data storage system that includes: a memory controller with one or more processors; a non-volatile memory with a datastore and a log stream; and a key-map, wherein the key-map stores datastore location information for a plurality of keys, the plurality of keys corresponding to data objects in one or more tiered data structures stored in the datastore; the method comprising:
>
> at the memory controller operatively coupled with the non-volatile memory and the key-map:
>
> in response to receiving a request, from a requestor, to perform a transaction including a set of two or more memory operations to be performed on the one or more tiered data structures, wherein a first memory operation in the set of two or more memory operations is associated with a first key:
>
> *writing a start transaction record to the log stream including a unique transaction identifier corresponding to the transaction*; and
>
> performing the two or more memory operations, including:
>
> *for each of the memory operations, upon completion of the memory operation, writing an operation*

[1] We use the word "Appellant" to refer to "applicant" as defined in 37 C.F.R. § 1.42. According to Appellant, SanDisk Technologies LLC is the real party in interest. Appeal Br. 3.

> *commit record to the log stream*, wherein the operation commit record guarantees a memory operation level of atomicity across the one or more tiered data structures; and
>
> for the first memory operation in the set of two or more memory operations:
>
> writing a new data object in the datastore;
>
> assigning, in the key-map, a location of the new data object in the datastore to the first key; and
>
> maintaining an old data object in the datastore, wherein a location of the old data object was previously assigned, in the key-map, to the first key;
>
> wherein the operation commit record for the first memory operation includes the first key, the unique transaction identifier, a pointer to the location of the new data object in the datastore, and a pointer to the location of the old data object in the datastore; and
>
> in accordance with a determination that the transaction, including the set of two or more memory operations in the transaction, is complete, *writing a transaction commit record to the log stream including the unique transaction identifier* corresponding to the transaction, wherein the transaction commit record guarantees a transaction level of atomicity across the data storage system.

See Appeal Br. 21 (emphases added).

REJECTIONS

Claims 1–13, 21–23, and 25 stand rejected under 35 U.S.C. § 103 as being unpatentable over Adl-Tabatabai et al. (US 2011/0145512 A1; June 16, 2011) ("Adl-Tabatabai"), Marathe et al. (US 2011/0252067 A1; October 13, 2011) ("Marathe"), and Lomet et al. (US 2012/0179645 A1; July 12, 2012) ("Lomet"). Final Act. 3; Ans. 3.

Claims 14–20 and 24 stand rejected under 35 U.S.C. § 103 as being unpatentable over Adl-Tabatabai, Marathe, Walsh et al. (US 9,170,938 B1; October 27, 2015) ("Walsh"), and Lomet. Final Act. 26; Ans. 3.

ANALYSIS

We have only considered those arguments that Appellant actually raised in the Briefs.[2] Arguments Appellant could have made, but chose not to make, in the Briefs have not been considered and are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(iv) (2019).

A. Whether the Examiner erred in finding that Lomet teaches "writing a start transaction record to the log stream"

The Examiner finds that Lomet teaches, *inter alia*, a transaction component that writes an operation, its log sequence number, and information from a reply message to its log. Final Act. 7 (citing Lomet at Abstract). The Examiner finds, in particular, that "every operation of a transaction is associated with a new log sequence number." Ans. 5.

The Appellant contends that the writing a transaction record to the log stream of Lomet, contrary to the Examiner's finding, is not a "start" transaction. Appeal Br. 13–14. The step of writing to the log, Appellant argues, occurs after completion of the operations and after completion of the transaction. *Id.* at 14; Reply Br. 5. More specifically, Appellant asserts, Lomet at figure 3 "teaches a 'write operation and log sequence number to

[2] Claims 2 and 7–24 are not argued separately from claim 1 in either of Appellant's briefs (Appeal Br. 12–20; Reply Br. 5–11), and will not be separately addressed.

log' at step 334 . . . that occur[s] **after completion** of the transaction." Reply Br. 5.

Our reviewing court has held that "[t]here is a 'heavy presumption' that the terms used in claims 'mean what they say and have the ordinary meaning that would be attributed to those words by persons skilled in the relevant art.'" *SuperGuide Corp. v. DirecTV Enters., Inc.*, 358 F.3d 870, 874–75 (Fed. Cir. 2004) (citing *Tex. Digital Sys., Inc. v. Telegenix, Inc.*, 308 F.3d 1193, 1202 (Fed. Cir. 2002)). Although claim terms are interpreted in light of the specification, we do not read limitations from the specification into the claims. *See Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571–72 (Fed. Cir. 1988) (various limitations on which appellant relied were not stated in the claims; the specification did not provide evidence indicating these limitations must be read into the claims to give meaning to the disputed terms); *see also In re Am. Acad. Of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004). In addition, proper claim construction requires a broadest reasonable interpretation consistent with the specification. *In re Bond*, 910 F.2d 831, 833 (Fed. Cir. 1990); *see also Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005).

Per the guidance above, we find that Appellant's argument is not commensurate with the scope of claim 1. Applying a broad but reasonable interpretation of the claim language in view of Appellant's Specification, we disagree with the Appellant that the limitation at issue—"writing a start transaction record to the log stream"—requires that the start transaction record must be written to the log stream prior to completion of the transaction.

Appellant's Specification describes "start transaction record" as being a record that "includ[es] a unique transaction identifier corresponding to the transaction." Spec. ¶ 19. A person having ordinary skill in the art at the time of Appellant's invention would not, under broadest reasonable interpretation, interpret Appellant's "start transaction record" as being required to be written either at the start or prior to completion of the transaction.

Moreover, Lomet teaches a step 322 of "determine log sequence number," which occurs for each operation of a transaction. *See* Lomet fig. 3. Lomet teaches that in this step 322, "[a] log sequence number may be determined in block 322 for the transaction." Lomet ¶ 67. The Examiner indicates that this issuance of a "log sequence number" is the start transaction record of Lomet. Ans. 5-6. According to figure 3 of Lomet, step 322 occurs at the start of the transaction. *See* Lomet, fig. 3. Therefore, even were we to construe the language of claim 1 to require the transaction record to be at the start of the transaction, we would find such a teaching in Lomet.

B. Whether Lomet teaches "a unique transaction identifier corresponding to the transaction" that is included in the "start transaction record"

The Examiner finds that Lomet teaches "a unique transaction identifier corresponding to the transaction," in the form of a new log sequence number. Final Act. 7 (emphasis omitted). The Examiner explains:

> [w]hen a new transaction starts, according to ¶0079, it starts with a new sequence number starting after the end of previous transaction. For example, if first transaction uses sequence

> numbers 1-5, the second transaction will start with sequence number 6 and so on. Therefore, when a transaction ends and it writes its last sequence number which is unique compared to subsequent transactions' sequence number to be written in the log.

Id.

Appellant disagrees with the Examiner's finding, arguing that the "log sequence number" of Lomet is not a "unique transaction identifier corresponding to the transaction," since a "log sequence number" is determined for each operation, rather than for each transaction. Appeal Br. 14–15. The Appellant points to the Abstract of Lomet, and argues: "The transaction component may manage a transaction using a log sequence number for each operation. . . . The transaction component may then write the operation, its log sequence number, and information from the reply message to its log." *Id.* (emphases omitted); *see also* Reply Br. 6–8.

Similar to Appellant's arguments presented *supra*, at section A, we find that the Appellant's arguments are not commensurate with the scope of the claims.

A broadest reasonable interpretation of "a unique transaction identifier corresponding to the transaction" does not preclude multiple unique transaction identifiers corresponding to each transaction. The Examiner, citing to the Abstract and paragraph 12 of Lomet for teaching of "monotonically increasing log sequence [number]" for each operation, finds that

> if a transaction had 5 operations, the log sequence numbers for those 5 operations can be 10, 11, 12, 13[,] and 14. Consider, the next transaction have [*sic*] three operations. Log sequence numbers of those operations will be 15, 16[,] and 17. So, we see

> *First transaction starts with a log sequence number 10 and the second transaction starts with a log sequence number 15.*

Ans. 5 (emphasis added). We agree with the Examiner that each transaction has a unique transaction identifier. Nor does the Appellant apprise us of any Examiner error in finding that the unique transaction identifier is included in the "start transaction record."

Accordingly, we sustain the Examiner's obviousness rejections of claims 1, 2, and 7–24.

C. Whether Marathe teaches "allocating a first slab" of a predefined size[3] to write the new data[4], per claim 3

The Examiner finds that Marathe teaches, *inter alia*, "each slab…has a respective size, of a set of predefined sizes," because Marathe teaches detecting a sector size of a persistent storage, and ensuring that each node "is not larger than the sector size of the persistent storage." Final Act. 9 (citing Marathe ¶ 3). The Examiner further finds that Marathe teaches the limitation "prior to writing the new data object in the datastore, allocating the first slab in the datastore for the new data object," because Marathe teaches "[a]t operation 408, the sector containing the newly inserted entry can be flushed to persistent storage." Final Act. 10 (emphasis omitted, citing Marathe ¶ 48).

[3] A limitation "each slab of the plurality of slabs has a respective size, of a set of predefined sizes," appears in claim 2 (Appeal Br. 22), from which claim 3 depends. *Id.*

[4] Claims 4–6 are not argued separately from claim 3 in either of Appellant's briefs (Appeal Br. 12–20; Reply Br. 5–11), and will not be separately addressed.

The Examiner finds, in particular, that memory must be allocated to flush data in a storage. *Id.*

Appellant argues "Marathe merely teaches 'receiving a key . . . [and] searching a B+ data structure using the key to find a leaf node." Appeal Br. 17 (alteration in original, no citation to Marathe provided). Appellant further argues that Marathe teaches only that each respective B+ tree node may have a smaller or similar size as the sector size, and thus Marathe does not teach whether each B+ tree node is of a predefined size. Reply Br. 9. Furthermore, the Appellant asserts that the Examiner's finding regarding flushing a newly inserted entry is inconsistent with the Examiner's finding that the nodes of Marathe are the claimed slabs; and that the flushing to persistent storage occurs after new data has been written in the node. Appeal Br. 18. Appellant additionally disputes the Examiner's finding that memory must be allocated to flush data in a storage. *Id.*

1. *Whether Marathe teaches "each slab . . . has a respective size, of a set of predefined sizes"*

We disagree with Appellant's argument that Marathe does not teach whether each B+ tree node is of a predetermined size.

In addition to the roadmap discussed above, at § A, our reviewing courts have held that a broadest reasonable interpretation of the claim language may involve extrinsic evidence such as "the appropriate use of dictionaries" for "assist[ing] in understanding the commonly understood meaning of words." *Phillips*, 415 F.3d at 1322 (citing *Exhibit Supply Co. v. Ace Patents Corp.*, 315 U.S. 126, 134 (1942) (relying on dictionaries to construe the claim term "embedded"); *Weber Elec. Co. v. E.H. Freeman*

Elec. Co., 256 U.S. 668, 678 (1921) (approving circuit court's use of dictionary definitions to define claim terms); *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1247–53 (Fed. Cir. 1998) (approving the use of dictionaries with proper respect for the role of intrinsic evidence)). The *Phillips* court held that, unless intrinsic evidence such as the specification or prosecution history provides otherwise, or the claim language is a term of art, dictionary use to define claim terms is permissible. *Phillips*, 415 F.3d at 1321–22.

Here, since Appellant's Specification does not provide a special definition for "predetermined," and the term is not one of art, we turn to dictionaries to provide a definition for "predetermined." Merriam-Webster Dictionary defines "predetermine" as "to determine beforehand" or "to impose a direction or tendency on beforehand." *Predetermine*, MERRIAM-WEBSTER, https://www.merriam-webster.com/dictionary/predetermine (last visited Mar. 9, 2020). The Cambridge English Dictionary provides a similar "decided or arranged at an earlier time" as the definition of "predetermined." *Predetermined*, CAMBRIDGE DICTIONARY, https://dictionary.cambridge.org/us/dictionary/english/predetermined (last visited Mar. 9, 2020).

As such, we find that the cited portions of Marathe, on which the Examiner relies, teach "each slab of the plurality of slabs has a respective size, of a set of predefined sizes," per the language of claim 2. Appeal Br. 22. Specifically, we agree with the Examiner that Marathe's teaching that each node "is not larger than (or is the same size as) the sector size of the persistent storage" (Ans. 6 (emphasis omitted, citing Marathe ¶ 63)) is a teaching for a set of sizes that are determined beforehand, i.e. "predetermined."

2. *Whether Marathe teaches "prior to writing the new data object in the datastore, allocating the first slab . . . for the new data object"*

We are not persuaded by Appellant's arguments that the Examiner erred in finding that Marathe teaches the above limitations.

In response to the Appellant's first argument that the Examiner's discussion "concerning flushing a newly inserted entry to persistent storage . . . is not consistent with the Examiner's assertion on page 9 of the [final] office action that Marathe's nodes are the equivalent of slabs" (Appeal Br. 18), the Examiner elucidates:

> Examiner refers to ¶0048 of Marathe: "At operation 408, the sector containing the newly inserted entry can be **flushed to persistent storage.",** and ¶0052 "For example in an embodiment, **a flush to disk of the free space management structure can be initiated after allocation** of eight blocks" Each sector containing a node of the B+ tree is synonymous to a slab of predefined size. Those sectors are flashed [*sic*] in the persistent storage/disk.

Ans. 6. Appellant's Reply Brief does not repeat the assertion that the flushing of a newly inserted entry is inconsistent with the Examiner's finding that the nodes of Marathe are analogous to the claimed slabs.

We do not find inconsistency with the Examiner's determination that Marathe teaches each sector containing a node, and the sector can be flushed to persistent storage, based on the support the Examiner provides for Marathe teaching each sector, which contains a node, being flushed to persistent storage. *Id.*

Regarding Appellant's additional arguments: the Examiner's findings that "[t]o flush the data in storage, memory need [*sic*] to be allocated" (Final

Act. 10); and "allocating the first slab" occurs "prior to writing the new data object" each amount to an assertion of inherency, without providing sufficient support (Appeal Br. 18 (emphasis omitted)), we disagree for reasons explained below.

The Examiner cites to Marathe paragraph 52 for teaching "that flushing of data in disk requires allocation of free disk space." Ans. 6. Moreover, Marathe at paragraph 52 teaches the claimed allocating occurring prior to the claimed writing. *Id.* Specifically, we agree with the Examiner that "a flush to disk . . . can be initiated after allocation" (Marathe ¶ 52) is a teaching of the claim limitation "prior to writing the new data object in the datastore, allocating the first slab . . . for the new data object." Appeal Br. 22. In other words, Marathe teaches allocating the first slab, then a flush to disk (Marathe ¶ 52), and finally writing the new data object in the datastore. *Id.* and Marathe ¶ 48 ("[i]n this fashion *a new entry can be inserted into a B+ tree data structure*" (emphasis added)).

Therefore, we sustain the Examiner's obviousness rejections of claims 3–6.

D. Whether Lomet teaches "identifying an incomplete transaction," wherein "the incomplete transaction includes a start transaction record having a unique transaction identifier in the log stream," and "performing a recovery process for the incomplete transaction, wherein . . . [the] incomplete transaction . . . includes a start transaction record having a lowest sequence number in the log stream"[5]

The Examiner finds that Lomet, at paragraph 56, teaches the limitations of claim 25. Final Act. 25–26.

Appellant argues that Lomet does not teach a "start transaction commit record," and thus cannot teach either "the incomplete transaction includes a start transaction record having a unique transaction identifier in the log stream," or "a start transaction commit record . . . without a corresponding transaction commit record," per the language of claim 25. Appeal Br. 20. The Appellant further argues that, since Lomet lacks a start transaction commit record without a corresponding transaction commit record, Lomet does not teach "identifying an incomplete transaction" and "performing a recovery process" based on the start transaction commit record as claimed. *Id.* Appellant further argues that, contrary to the claimed features, Lomet "merely performs a recovery from a 'redo scan start point,' to redo 'all of the operations with log sequence numbers greater than the redo scan start point.'" Reply Br. 11 (citing Lomet ¶ 86); *see also* Appeal Br. 20.

[5] This limitation appears in claim 25. Appeal Br. 30.

Appellant's arguments challenging the Examiner's finding that Lomet teaches "a start transaction record having a unique transaction identifier in the log stream" have already been discussed, *supra*, at section A, and will not be further addressed.

In response to the Appellant's argument that Lomet does not teach "identifying an incomplete transaction" and "performing a recovery process," the Examiner finds:

> Fig. 5 of Lomet . . . clearly shows the recovery form [*sic*] an erroneous (i.e., incomplete) transection [*sic*] using the log sequence number. ¶0086 to ¶0088 explains the steps of the recovery from an incomplete transaction. Examiner cites from ¶0086 to clarify the recovery [redo] operation: "Embodiment 500 illustrates a method that may be performed to recover from an error with a transaction component. The error may cause the transaction component to be restarted, which may lose any information that has not been saved. *In order to effectively resume, a redo scan start point may be used to begin recovery from a log. . . . So, when an error occurs in a transaction, to recover from the error, a redo scan is done thought [sic] the log to find out the start of the transaction and those incomplete operations are re-sent.*

Ans. 7 (original emphases removed; emphasis added).

We agree with the Examiner that figure 5 and the cited portions of Lomet teach the claim language at issue. Specifically, Lomet at step 502 teaches "identifying an incomplete transaction in the log stream." *See* Lomet at figure 5; Ans. 7. Lomet at paragraphs 56 and 86 teaches "wherein the incomplete transaction . . . [is] without a corresponding transaction commit record"; more specifically, we agree with the Examiner that Lomet teaches that the incomplete transaction has not been added to the "stable log." Final Act. 25–26. We further agree with the Examiner's finding that Lomet shows "performing a recovery process" in at least figure 5. Ans. 7. Further, we agree with the Examiner that Lomet teaches "wherein the earliest incomplete transaction . . . includes a start transaction record having a lowest sequence number in the log stream," at least in part because the Examiner finds that Lomet teaches "monotonically increasing log sequence numbers." Lomet ¶ 12; Final Act. 25–26; Ans. 6–7. Therefore, upon detection of an error, Lomet teaches a redo to a previous, lower-number log sequence number. Lomet ¶ 12; *see also* Lomet fig. 5.

Accordingly, we sustain the Examiner's obviousness rejection of claim 25.

DECISION

For at least the reasons described *supra*:

We affirm the Examiner’s decision rejecting claims 1–13, 21–23, and 25 under 35 U.S.C. § 103 as being unpatentable over Adl-Tabatabai, Marathe, and Lomet.

We affirm the Examiner’s decision rejecting claims 14–20 and 24 under 35 U.S.C. § 103 as being unpatentable over Adl-Tabatabai, Marathe, Walsh, and Lomet.

CONCLUSION

Claim(s) Rejected	**35 U.S.C. §**	**Reference(s)/Basis**	**Affirmed**	**Reversed**
1–13, 21–23, 25	103	Adl-Tabatabai, Marathe, Lomet	1–13, 21–23, 25	
14–20, 24	103	Adl-Tabatabai, Marathe, Walsh, Lomet	14–20, 24	
Overall Outcome			1–25	

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv). *See* 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED