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.
15/058,262	03/02/2016	Idan Burstein	1058-1196	4701

82253 7590 06/22/2020

KLIGLER & ASSOCIATES PATENT ATTORNEYS LTD.
P.O. Box 57651
Tel Aviv, 61576
ISRAEL

EXAMINER	
PADMANABHAN, MANO	

ART UNIT	PAPER NUMBER
2138	

NOTIFICATION DATE	DELIVERY MODE
06/22/2020	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):

alon@dkpat.co.il
daniel@dkpat.co.il
info@dkpat.co.il

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte IDAN BURSTEIN, DIEGO CRUPNICOFF, SHLOMO RAIKIN, and MICHAEL KAGAN

Appeal 2018-007186
Application 15/058,262
Technology Center 2100

Before JOHNNY A. KUMAR, JOHN A. EVANS, and JOHN P. PINKERTON, *Administrative Patent Judges*.

PINKERTON, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellant[1] appeals under 35 U.S.C. § 134(a) from the Examiner's Final Rejection of claims 1–9, 12–33, and 36–48, which are all of the claims pending in the application. Claims 10, 11, 34, and 35 are canceled. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.

[1] We use the word "Appellant" to refer to "applicant" as defined in 37 C.F.R. § 1.42. Appellant identifies Mellanox Technologies Ltd. as the real party in interest. Appeal Br. 1.

STATEMENT OF THE CASE

Introduction

Appellant generally describes the disclosed and claimed invention as follows:

> A memory device includes a target memory, having a memory address space, and a volatile buffer memory, which is coupled to receive data written over a bus to the memory device for storage in specified addresses within the memory address space. A memory controller is configured to receive, via the bus, a flush instruction and, in response to the flush instruction, to immediately flush the data held in the buffer memory with specified addresses within the memory address space to the target memory.

Abstract.[2]

Claims 1, 15, 25, and 39 are independent. Claim 1 is illustrative of the subject matter on appeal and is reproduced below:

> 1. Computing apparatus, comprising:
>
> a memory device, comprising:
>
> a target memory, having a memory address space;
>
> a volatile buffer memory, which is coupled to receive data written over a bus to the memory device for storage in specified addresses within the memory address space; and
>
> a memory controller, which is configured to receive, via the bus, a flush instruction and, in response to the flush instruction, to immediately flush the data held in the

[2] Our Decision refers to the Final Office Action mailed Sept. 29, 2017 ("Final Act."); the Appeal Brief filed Feb. 19, 2018 ("Appeal Br."); the Reply Brief filed July 3, 2018 ("Reply Br."); the Examiner's Answer mailed May 18, 2018 ("Ans."); and the original Specification filed Mar. 2, 2016 ("Spec.").

buffer memory with specified addresses within the memory address space to the target memory; and

a network interface controller (NIC), which is coupled to receive from a peer node, via a network, a remote direct memory access (RDMA) write packet containing the data to be written to the memory device and specifying the addresses to which the data are to be written, and to receive an RDMA flush packet, and which is configured, in response to the RDMA write and flush packets, to convey the data and the flush instruction over the bus to the memory device, without generating an interrupt to a central processing unit (CPU) that is coupled to the bus in the apparatus.

Appeal Br. 27 (Claims App.).

References

Name	Publication Number	Date
Kriz	US 4,680,703	July 14, 1987
Batchelor et al. ("Batchelor")	US 6,490,647 B1	Dec. 3, 2002
Feng et al. ("Feng")	US 2006/0256784 A1	Nov. 16, 2006
Ehrlich et al. ("Ehrlich")	US 2010/0205367 A1	Aug. 12, 2010
Hong et al. ("Hong")	US 2014/0258637 A1	Sept. 11, 2014
Das	US 2015/0067091 A1	Mar. 5, 2015
Mukundan et al. ("Mukundan")	US 9,104,582 B1	Aug. 11, 2015
Govind	US 2017/0034268 A1	Feb. 2, 2017
Voigt	WO 2014158168 A1	Oct. 2, 2014
PCI Express	PCI Express® Base Specification, Rev. 3.0	Nov. 10, 2010

Rejections on Appeal[3]

Claims 1–3, 6, 8, 9, 12, 13, 25–27, 30, 32, 33, 36, and 37 stand rejected under 35 U.S.C. § 103 over Ehrlich, Feng, and Das.

Claims 4 and 28 stand rejected under 35 U.S.C. § 103 over Ehrlich, Feng, Das, and Voigt.

Claims 5 and 29 stand rejected under 35 U.S.C. § 103 over Ehrlich, Feng, Das, Voigt, and Hong.

Claims 7 and 31 stand rejected under 35 U.S.C. § 103 over Ehrlich, Feng, Das, Batchelor, Voigt, and Kriz.

Claims 14 and 38 stand rejected under 35 U.S.C. § 103 over Ehrlich, Feng, Das, and Govind.

Claims 15–17 and 39–41 stand rejected under 35 U.S.C. § 103 over Ehrlich and Mukundan.

Claims 18 and 42 stand rejected under 35 U.S.C. § 103 over Ehrlich, Mukundan, and Das.

Claims 20–23 and 44–47 stand rejected under 35 U.S.C. § 103 over Ehrlich, Mukundan, Feng, and Das.

Claims 24 and 48 stand rejected under 35 U.S.C. § 103 over Ehrlich, Mukundan, Feng, Das, and Govind.

Claims 19 and 43 stand rejected under 35 U.S.C. § 103 over Ehrlich, Mukundan, and PCI Express.

ANALYSIS

Claims 1–9, 12–14, 25–33, and 36–38

[3] The Leahy-Smith America Invents Act ("AIA") included revisions to 35 U.S.C. § 100 *et seq.* effective on March 16, 2013. Because this application was filed after March 16, 2013, the Examiner examined the claims under the AIA version of 35 U.S.C. § 103. Final Act. 2.

The Examiner rejects independent claims 1 and 25 under § 103 as being unpatentable over Ehrlich, Feng, and Das. Final Act. 3–6, 10–13.

The dispositive issue raised by Appellant's briefs is whether the combination of Ehrlich, Feng, and Das teaches or suggests a network interface controller (NIC), which is coupled to receive from a peer node, via a network, "an RDMA flush packet, and which is configured, in response to the RDMA . . . flush packet[], to convey the . . . flush instruction over the bus to the memory device . . . ," as recited in claim 1 and as similarly recited in claim 25.[4]

In the Final Action, the Examiner finds that Ehrlich and Feng do not appear to recite an RDMA flush packet. Final Act. 5. However, the Examiner finds that Das teaches:

> To receive an RDMA flush packet, and which is configured, in response to the RDMA write and flush packets, to convey the data and the flush instruction over the bus to the memory device *(see Das, [0048] where the flush request is received and the write data and the flush packet containing the flush request are sent to the data storage device).*

Id. at 6.

In the Answer, the Examiner finds that although claim 1 states "receive an RDMA flush packet" and "RDMA write and flush packets," "nowhere in the quoted areas or in the claim, does it state that the flush packet is a dedicated RDMA flush packet that is separate." Ans. 32. Thus, the Examiner finds that "any prior art [that] teaches an RDMA flush packet, even if the packet is not solely just for a flush and includes any other data or

[4] Appellant argues claims 1 and 25 together. Appeal Br. 5–12. Thus, we select claim 1 as representative with claim 25 standing or falling with claim 1. *See* 37 C.F.R. § 41.37(c)(1)(iv).

functions, . . . would read on the recited claim." *Id.* The Examiner also finds that "Feng and Das both teach an RDMA system for transmitting data remotely" and that Feng teaches RDMA in paragraph 47 and a flush packet in paragraph 97 "as a flush command that is sent, through the TCP layer." *Id.* The Examiner further finds that Das teaches "the flush packets are sent to the remote node as a part of the set of write requests, therefore the flush packet is sent on behalf of the local node . . . [and] then the data is flushed to the memory device from the remote node." *Id.* Moreover, the Examiner finds that "[i]n further support of this in Das [0048] the remote node even sends back an acknowledgement that the flush has been completed in response to the request." *Id.*

Appellant argues that none of the cited references, individually or in any combination, teaches or suggests "anything that even remotely resembles an RDMS flush packet." Appeal Br. 6 (emphasis omitted); Reply Br. 2 (emphasis omitted). Appellant also argues that claims 1 and 25 specifically define "an RDMA flush packet as a separate and distinct entity, with a well-defined purpose" and that the Examiner has applied an unreasonably broad interpretation of claims 1 and 25 by apparently taking the position "that any packet that is followed by flushing of a memory is an 'RDMA flush packet.'" Reply Br. 3 (emphasis omitted).

We are persuaded by Appellant's arguments that the Examiner erred. First, we agree with Appellant that the recited "RDMA flush packet" in claims 1 and 25 is a separate packet having the specific purpose of causing the NIC to convey the flush instruction over the bus to the memory device. For example, claim 1 expressly recites that an NIC is coupled to receive from a peer node "a remote direct memory access (RDMA) write packet"

and "an RDMA flush packet." Claim 1 also recites that the NIC "is configured, in response to the RDMA write and flush packets, to convey the data and the flush instruction over the bus to the memory device." The descriptions in the Specification fully support that the RDMA flush packet is a separate packet that causes the NIC to convey the flush instruction. *See, e.g.*, Spec. 3:17–25; 4:3–11. Based on the language of claims 1 and 25, and the descriptions of the RDMA flush packet in the Specification, we agree with Appellant that the Examiner's interpretation of the "RDMA flush packet" is overly broad, unreasonable, and inconsistent with the Specification. The Examiner's claim interpretation may be the broadest possible interpretation, but it is not the broadest *reasonable* interpretation consistent with the Specification. *See In re Smith Int'l, Inc.*, 871 F.3d 1375, 1382–83 (Fed. Cir. 2017) (The "correct inquiry" is to give a claim its broadest *reasonable* interpretation, not the "broadest *possible* interpretation.").

Second, we agree with Appellant's argument that the references cited by the Examiner do not teach or suggest the RDMA flush packet, either individually or collectively. Appellant argues that "Feng is directed to resolving inefficiencies in use of I/O resources and existing RDMA protocols, by using a simplified RDMA NIC architecture." Appeal Br. 8 (citing Feng ¶¶ 4–7). Appellant notes, but does not dispute, that the Examiner "cited Feng as an example of conventional RDMA write functionality (paragraph 0046), along with the fact that by definition in RDMA, the memory is accessed without interrupting the CPU (paragraph 0047)." Appeal Br. 8. Although we agree with the Examiner that Feng teaches RDMA functionality and using "an RNIC in the remote system" to

allow accessing memory without interrupting the CPU, the Examiner has not explained, and we do not discern, how paragraph 97 of Feng teaches or suggests an "RDMA flush packet." *See* Ans. 32; Feng ¶ 47.

We also agree with Appellant that Das does not teach or suggest an NIC coupled to receive from a peer node "an RDMA flush packet," and configured, in response to the RDMA flush packet, to convey the flush instruction over the bus to the memory device. Appellant argues, and we agree, that "[t]he method described by Das is dependent on a processor on the node that receives the RDMA packets to 'execute program instructions to perform embodiments of the present invention.'" Appeal Br. 8 (citing Das ¶ 28). Paragraph 13 of Das explains that:

> The processor receives one or more data write requests through multiple independent pathways, executes the one or more data write requests to a cache and flushes newly written data to the data storage device. . . . The processor then sends an acknowledgement that the one or more data write requests have been flushed and waits to receive a metadata write request related to the one or more data write requests.

Thus, as Appellant argues, and we agree, "the node that sent the RDMA packets containing the data to be written has no direct involvement in the flush operation and must simply wait for this delayed acknowledgment." Appeal Br. 9.

Appellant argues that this mode of operation is also illustrated in Figure 5 of Das and described in paragraph 48 of Das, which is cited by the Examiner. *Id.* In that regard, Appellant argues that as described in paragraph 48, "a 'local node' sends first and second sets of data write requests 504 and 506 to a 'remote node,' and the remote node then sends 'a first acknowledgement 508 that the first set of data write requests 504 was

flushed to a data storage device . . . and a second acknowledgment 510 that the second set of data write requests 506 was flushed a data storage device.'" *Id.* at 9–10. Appellant further argues that as illustrated by the time line in Figure 5, the first acknowledgement 508 reaches the local node before the local node sends a "metadata write and flush request 514," and the same is true with respect to the second acknowledgment 510 and the second "metadata write request 516." *Id.* at 10. Based on Figure 5 and paragraph 48 of Das, we agree with Appellant's arguments. Thus, we also agree with Appellant's argument that "Das's 'remote node' flushes data to a storage device without waiting for any sort of RDMA flush request from the 'local node,'" and Das's '"metadata write and flush request' has nothing to do with flushing of previously-written RDMA data." *Id.*

In view of the foregoing, we do not sustain the Examiner's rejection of independent claims 1 and 25 under 35 U.S.C. § 103. For the same reasons, we do not sustain the Examiner's rejection of dependent claims 2–9, 12–14, 26–33, and 36–38. *Cf. In re Fritch*, 972 F.2d 1260, 1266 (Fed. Cir. 1992) ("dependent claims are nonobvious if the independent claims from which they depend are nonobvious").

Claims 15–24 and 39–48

The Examiner rejects claims 15 and 39 under 35 U.S.C. § 103 over Ehrlich and Mukundan. Final Act. 22–26. Appellant argues that the references cited by the Examiner, taken individually or in combination, do not teach or suggest "the use of a flag that is set in the header of a transaction packet to indicate data to be flushed in response to a subsequent flush instruction, as recited in claims 15 and 39." Appeal Br. 12 (emphasis omitted); Reply Br. 4. Appellant also argues that the cited references do not

teach or suggest "any sort of selective flush of data written to a memory device by another device on a bus, let alone the use of a flag in a packet header for this purpose." Appeal Br. 13 (emphasis omitted).

In the Answer, the Examiner finds that "any prior art which contains a header with a transaction descriptor, which comprises a flush flag which is set in the transaction packet, a payload, and then further flushes in response to a flush instruction will read on the recited claim." Ans. 34. In that regard, the Examiner also finds that Ehrlich teaches the data received (which is the transaction packet) encompasses metadata (which is the transaction descriptor), "which includes a dirty flag (flush flag)." *Id.* at 35. The Examiner further finds that in Ehrlich, the memory controller is taught and the flush command is received, and "the flush instruction causes the memory controller to immediately flush the data due to the dirty flag being set for each data entry." *Id.* at 35–36 (citing Ehrlich ¶¶ 20–24, 28). According to the Examiner, Mukundan teaches that "metadata (which is the transaction descriptor from Ehrlich, which includes the dirty flag) is stored in the header of the packet." *Id.* at 35 (citing Mukundan 5:41–49).

We are persuaded by Appellant's arguments that the Examiner erred. First, Appellant argues, and we agree, that "Ehrlich does not refer to transaction packets, and thus cannot be taken to teach that the packet headers comprise a transaction descriptor comprising a flush flag." Appeal Br. 14 (emphasis omitted). As Appellant explains, Ehrlich describes techniques to maintain data integrity between a write-cache and flush-cache location of a disk drive, to prevent stale data in the flush-cache location from being restored into the write-cache upon power loss. *Id.* at 8 (citing Ehrlich ¶ 9), 13 (emphasis added). Each entry in Ehrlich's write-cache includes a dirty

flag (i.e., a bit), data to be written, and a location on the disk where the data is to be written. *Id.* at 13 (citing Ehrlich ¶ 20). Appellant argues, and we agree, that Ehrlich's dirty flag is set whenever the disk drive receives a write command from the host computer, and Ehrlich does not teach or suggest that the host computer might have any discretion in determining whether to set the dirty flag. *Id.* at 14 (citing Ehrlich Fig. 4, ¶ 27). When the disk drive receives a flush-cache command, all entries in the write-cache having their dirty flags set are written to a flush-cache location. *Id.* (citing Ehrlich ¶ 28).

Second, Appellant argues, and we agree, that Mukundan teaches "the well-known fact" that a packet header may include metadata of a media file. *Id.* (citing Mukundan 5:41–42). Appellant candidly states, however, that although neither Ehrlich nor Mukundan teaches or suggests the use of a transaction descriptor in a packet header, the Specification "makes clear that transaction packets are used to carry data over the well-known PCI Express bus, and such packets include a transaction descriptor in their headers (page 15, first full paragraph, citing the *PCI Express Base Specification*)." *Id.* at 14–15.

Third, Appellant argues, and we agree, that the combination of Ehrlich and Mukundan, and even PCI Express, does not teach or suggest a flush flag in the header of a transaction packet that is used in writing data to a memory device. *Id.* at 15. As Appellant also argues, and as discussed above, the "dirty flag" described by Ehrlich "is set by the disk drive, i.e., by the memory device itself, for all data it receives from the host computer and writes to its cache." *Id.* Appellant further argues, and we agree, that "Ehrlich makes no suggestion that the dirty flag might be set selectively or

in response to some specific instruction, let alone in response to a flush flag in the header of a transaction packet, as required by claims 15 and 39." *Id.*

Thus, we agree with Appellant that the combination of the cited references does not teach or suggest (1) "each transaction packet comprising a header, which comprises a transaction descriptor, which comprises a flush flag, and a payload comprising the data" and (2) "in response to the flush instruction, to immediately flush to the target memory the data held in the buffer memory for which the flush flag was set in the transaction packets." *Id.* at 15–16; Reply Br. 4. We also agree with Appellant's argument that it is not enough for the Examiner to show merely that individual elements of claims 15 and 39 may be found in the cited references; instead, the Examiner must present a cogent rationale to explain why a person of ordinary skill would have had an apparent reason to select and combine the individual elements to arrive at the claimed invention. Reply Br. 4; *see In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) ("[R]ejections on obviousness grounds [require] some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.") (cited with approval in *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007)). The Examiner has failed to do so here.

For these reasons, we do not sustain the Examiner's rejection of independent claims 15 and 39 under 35 U.S.C. § 103. For the same reasons, we do not sustain the Examiner's rejection of dependent claims 16–24 and 40–48. *In re Fritch*, 972 F.2d at 1266.

DECISION

We reverse the Examiner's rejection of claims 1–9, 12–33, and 36–48 under 35 U.S.C. § 103.

SUMMARY

In summary:

Claims Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1–3, 6, 8, 9, 12, 13, 25–27, 30, 32, 33, 36, 37	103	Ehrlich, Feng, Das		1–3, 6, 8, 9, 12, 13, 25–27, 30, 32, 33, 36, 37
4, 28	103	Ehrlich, Feng, Das, Voigt		4, 28
5, 29	103	Ehrlich, Feng, Das, Voigt, Hong		4, 28
7, 31	103	Ehrlich, Feng, Das, Batchelor, Voigt, Kriz		7, 31
14, 38	103	Ehrlich, Feng, Das, Govind		14, 38
15–17, 39–41	103	Ehrlich, Mukundan		15–17, 39–41
18, 42	103	Ehrlich, Mukundan, Das		18, 42
20–23, 44–47	103	Ehrlich, Feng, Mukundan, Das		20–23, 44–47
24, 48	103	Ehrlich, Feng, Mukundan, Das, Govind		24, 48
19, 43	103	Ehrlich, Mukundan, PCI Express		19, 43
Overall Outcome				1–9, 12–33, 36–48

REVERSED