

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA CORPORATION,
Petitioner,

v.

POLARIS INNOVATIONS LIMITED,
Patent Owner.

Case IPR2017-01500
Patent 6,532,505 B1

Before MINN CHUNG, DANIEL J. GALLIGAN, and
JOHN A. HUDALLA, *Administrative Patent Judges*.

GALLIGAN, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
Inter Partes Review
35 U.S.C. § 318(a)

I. INTRODUCTION

In this *inter partes* review, NVIDIA Corporation (“Petitioner”) challenges the patentability of claims 1–18 of U.S. Patent No. 6,532,505 B1 (“the ’505 patent,” Ex. 1001), which is assigned to Polaris Innovations Limited (“Patent Owner”).

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision, issued pursuant to 35 U.S.C. § 318(a), addresses issues and arguments raised during the trial in this *inter partes* review. For the reasons discussed below, we determine that Petitioner has proven by a preponderance of the evidence that claims 1–18 of the ’505 patent are unpatentable. *See* 35 U.S.C. § 316(e) (“In an *inter partes* review instituted under this chapter, the petitioner shall have the burden of proving a proposition of unpatentability by a preponderance of the evidence.”).

A. Procedural History

On May 30, 2017, Petitioner filed a Petition (Paper 2 (“Pet.”)) requesting *inter partes* review of claims 1–18 of the ’505 patent based on the references below.

Ramanujan	US 5,202,973	Apr. 13, 1993	Ex. 1005
Hughes	US 5,784,582	July 21, 1998	Ex. 1008
Paluch	US 5,987,574	Nov. 16, 1999 (filed Apr. 30, 1997)	Ex. 1006
Chauvel	US 6,253,297 B1	June 26, 2001 (filed Oct. 13, 1998)	Ex. 1004

The grounds of unpatentability presented in the Petition are set forth in the table below.

Reference(s)	Basis	Claims
Chauvel	§ 103	1–18
Chauvel and Ramanujan	§ 103	3–13, 16, 18
Paluch and Hughes ¹	§ 103	1–7, 11–18
Paluch and Chauvel	§ 103	1–7, 11–18

Pet. 1–2. Patent Owner filed a Preliminary Response. Paper 7 (“Prelim. Resp.”).

We instituted trial as to claims 1–18 on the following grounds:

Whether claims 1–18 are unpatentable under 35 U.S.C. § 103(a) over the teachings of Chauvel;

Whether claims 3–13, 16, and 18 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Chauvel and Ramanujan;

Whether claims 1–4 and 11–18 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Hughes; and

Whether claims 1, 11–14, 16, and 17 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Chauvel.

Paper 8 (“Dec. on Inst.”), 31. We did not institute trial, however, on the following grounds:

¹ Petitioner initially identifies this ground as asserting obviousness “over Paluch in view of Hughes or Chauvel.” Pet. 1–2. We, however, understand this ground to be based on the combination of Paluch and Hughes only because Chauvel is relied upon in combination with Paluch in a separate ground. Pet. 4; *see also* Pet. 51 (“GROUND 3 & 4: Claims 1–7 AND 11–18 are obvious over Paluch in View of Hughes, and over Paluch in View of Chauvel”) (emphasis omitted).

Whether claims 5–7 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Hughes; and

Whether claims 2–7 and 15 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Chauvel.

Dec. on Inst. 28–30.

Following the Supreme Court’s decision in *SAS Institute Inc. v. Iancu*, 138 S. Ct. 1348 (2018), we modified the Decision on Institution to institute on all of the grounds presented in the Petition. Paper 17, 2. On May 10, 2018, the parties filed a joint paper acknowledging that the previously-denied grounds are now part of the trial and stating the following: “Patent Owner rests on its arguments made in its Response (Paper 15) because it believes that its arguments in the Response fully address both additional challenges, and the parties do not desire any changes in the schedule or any further briefing in this trial.” Paper 18, 1.

During the trial, Patent Owner filed a Response (Paper 15, “PO Resp.”), and Petitioner filed a Reply (Paper 19, “Pet. Reply”). With our authorization (Paper 20), Patent Owner filed a sur-reply addressing an issue of claim construction (Paper 21), and Petitioner filed a responsive brief (Paper 22). Patent Owner also filed a Motion to Exclude Exhibit 1012 (Paper 25), to which Petitioner filed an opposition (Paper 26) and in support of which Patent Owner filed a reply (Paper 28).

An oral hearing was held on August 17, 2018, a transcript of which appears in the record. Paper 29 (“Tr.”).

B. Real Parties in Interest

Patent Owner identifies itself, Wi-LAN Inc., and Quarterhill Inc. as real parties-in-interest. Paper 6, 2.

C. Related Matters

Petitioner and Patent Owner cite the following judicial matter involving the '505 patent: *Polaris Innovations Ltd. v. Dell Inc. & NVIDIA Corp.*, Case No. 4:16-cv-07005 (N.D. Cal.). Pet. 89; Paper 4, 2–3.

D. The '505 Patent and Illustrative Claim

The '505 patent generally relates to providing access to shared resources in a computing system using a “universal resource access controller.” Ex. 1001, Abstract, 1:6–11, 2:60–63. Claims 1 and 2 are illustrative and are reproduced below.

1. A universal resource access controller coupled to a requesting system and a resource, wherein when the requesting system desires access to the resource, the requesting system generates a resource access request which is passed to the universal resource controller which, in turn, uses a specific characteristic operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource to generate a corresponding sequenced command suitable for accessing the resource as required by the requesting system.
2. A universal resource access controller as recited in claim 1, wherein the universal resource controller comprises:
 - a configurable system interface coupled to the requesting system suitably arranged to both receive the resource access request and to generate a corresponding universal command;
 - a universal command sequencer coupled to the configurable system interface;
 - a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current

state of the requested resource and a resource tag arranged to identify the requested state of the requested resource; and

a characteristic operating parameter buffer coupled to the command sequencer arranged to store the characteristic operating parameter associated with the requested resource,

wherein the universal command sequencer uses the respective resource tags that identify the current state and the requested state of the requested resource and the characteristic operating parameter associated with the requested resource to generate the sequenced universal command.

II. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner's declarant, Dr. Tredennick, testifies that "a person of ordinary skill in the art as of the time of the '505 patent would have a Bachelor's degree in Electrical Engineering and at least 2 years of experience working in the field of semiconductor logic design." Ex. 1002

¶ 13. Neither Patent Owner, nor its declarant, Dr. Przybylski, offers a different assessment of the level of ordinary skill in the art. See Ex. 2002 ¶ 33 ("I am applying the definition of the level of experience of a person of ordinary skill in the art that has been put forward by Dr. Tredennick in his declaration (paragraph 13). I do not necessarily agree with the definition offered there, but I do not presently believe that the exact level of experience of a person of ordinary skill impacts my opinions offered herein.").

Based on the evidence of record, we adopt Dr. Tredennick's statement of the level of ordinary skill in the art with the exception of the language "at least." Thus, we determine that the skill level of a person of ordinary skill in the art would have been that of a person with a bachelor's degree in

electrical engineering and two years of experience working in the field of semiconductor logic design.

B. Claim Interpretation

In an *inter partes* review, claim terms in an unexpired patent are interpreted 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). In applying a broadest reasonable construction, claim terms generally are given their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *See In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). This presumption may be rebutted when a patentee, acting as a lexicographer, sets forth an alternate definition of a term in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

During the trial, the parties' dispute focuses on the broadest reasonable interpretation of "a resource tag buffer," as recited in claims 2 and 15.

1. "A resource tag buffer"

Claims 2 and 15 recite "a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current state of the requested resource and a resource tag arranged to identify the requested state of the requested resource."

Petitioner argues that, "[a]ccording to the specification of the '505 patent, the 'resource tag buffer' stores information about the current state and the requested state of the shared resource," and it contends, therefore, that the broadest reasonable interpretation of "resource tag is

simply a generic storage location capable of storing the claimed states.” Pet. 26 (citing Ex. 1001, 7:3–29, 7:46–60). In our Decision on Institution, we addressed Petitioner’s and Patent Owner’s arguments as to whether Chauvel and Hughes teach the claimed “resource tag buffer,” and we stated that “Petitioner has made a sufficient showing that the identified memory in Chauvel collectively teaches a ‘resource tag buffer,’ as recited in claim 2.” Dec. on Inst. 14; *see also id.* at 27 (same initial determination as to Hughes).

During the trial, the parties further developed the record on the broadest reasonable interpretation of “a resource tag buffer.” Patent Owner argues that the claim language “requires a resource tag buffer that stores two different types or categories of resource tags – one relating to a current state of the requested resource and one relating to the requested states of the resource.” PO Resp. 15. Patent Owner argues that the asserted prior art “disclos[es] multiple distinct storage structures” allegedly having the recited resource tags and that this is insufficient to teach “a resource tag buffer.” Paper 21, 1.

In its Reply, Petitioner cites cases in which the United States Court of Appeals for the Federal Circuit has determined that “a” in a claim means “one or more,” and Petitioner argues, therefore, that “a resource tag buffer” should not be limited to a single buffer. Pet. Reply 3–9. With our authorization (Paper 20), the parties filed additional papers addressing whether “a” in the term “a resource tag buffer” means “one or more.” Papers 21, 22. Patent Owner acknowledges that “there may be any number of resource tag buffers” but argues that “at least one must satisfy the entire limitation” to store both recited resource tags. Paper 21, 4.

Even if we resolved the parties' dispute as to whether "a" in the claim means "one or more," there still would be a question as to whether one of these buffers must still store both resource tags. More fundamental to the resolution of the issues in this case is determining the broadest reasonable interpretation of "buffer" and, more particularly, determining whether a buffer may comprise multiple storage structures or locations. Based on the evidence of record as explained below, we determine that a buffer may comprise multiple storage structures or locations.

We first turn to the specification of the '505 patent. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (en banc) ("[T]he specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.") (internal quotation and citation omitted). The written description of the '505 patent uses the phrase "resource tag buffer 114" or "tag buffer 114" six times, and two figures show a box labeled "resource tags 114." Ex. 1001, 7:30–37, 7:47–60, 8:65–9:2, 10:39–47, Figs. 1B, 1D. The '505 patent introduces the term in describing that command sequencer 116 provides a sequenced command "[u]sing the resource state information provided by the tags 300 stored in a resource tag buffer 114." Ex. 1001, 7:30–37. The '505 patent similarly states that "command sequencer 116 uses the information provided by LUT [(look-up table)] 118 in conjunction with the resource tags stored in the resource tag buffer 114 to properly sequence the command components 202–210 to form the sequenced command 220." Ex. 1001, 8:65–9:2. In another passage, the '505 patent states the following:

In another embodiment of the invention, when the shared resource 108 is a multi-bank type memory device, such as a SDRAM, or when the shared resource is a multi-device memory device such as a multi-chip module, the resource tag buffer 114 can store resource tags for all opened pages in a particular bank or device, for example. In one implementation, a comparator (not shown) detects a bank number or device identifier in the system address and compares the page address and the system address with the contents of the tag buffer 114. In the case where the comparison is not a “hit” (i.e., addresses don’t match), the universal controller 104 must close the old page using the address from the tag buffer 114 and open the new page based upon the new system command.

Ex. 1001, 7:47–60. These passages do not describe any particular structure for the resource tag buffer but, rather, describe the buffer in terms of what it stores, namely resource tags.

The figures of the ’505 patent do not provide any additional insight on the structure of the resource tag buffer. Figure 1B is reproduced below.

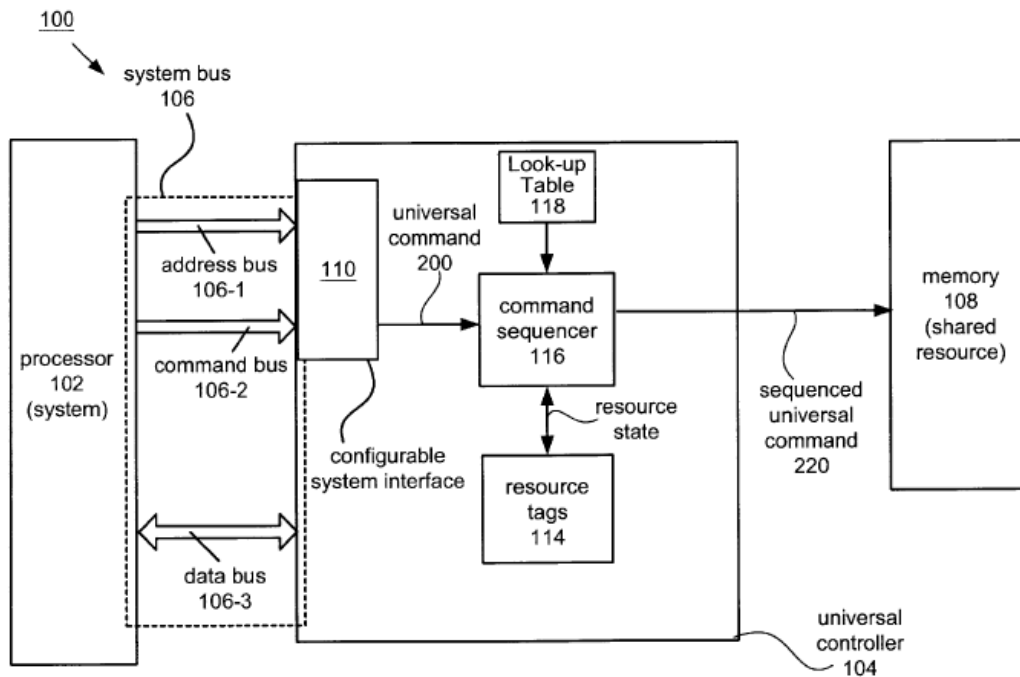


Fig. 1B

Figure 1B depicts an implementation of the universal controller of Figure 1A, and it depicts a box labeled “resource tags 114.” Ex. 1001, 3:34–35. Figure 1D includes the same box labeled “resource tags 114.” Patent Owner argues that these figures and other references in the ’505 patent “confirm that it [(resource tag buffer 114)] is the one place where all resource tags are stored.” PO Resp. 18 (citing Ex. 1001, 7:30–31, 7:50–52, 7:56–60, 8:67, 10:43). Patent Owner contends that “[n]o other storage location for the resource tags is mentioned or contemplated” and that “[n]othing in the specification indicates that different structures store different categories of resource tags (let alone storing a resource tag relating to the current state of the requested resource in one structure and a resource tag relating to the requested state of the requested resource in another, separate structure).” PO Resp. 18–19 (citing Ex. 2002 ¶ 47).

This line of argument raises the question of what it means for various pieces of data (resource tags) to be stored in “one place” or one “storage location.” The ’505 patent provides some insight on this question in the following passage:

[I]n order to track a state of the physical pages in the shared memory 108 having a number of memory banks, for example, a large number of resource tags which would require a large amount of cache memory dedicated to the resource tag buffer 114 are needed. This would slow the performance of the universal controller 104 since it would require substantial amounts of time to retrieve particular resource tags for particular pages of memory each of which may be located in disparate locations.

Ex. 1001, 10:39–47. This description of resource tags “requir[ing] substantial amounts of time to retrieve” because they may be stored in “disparate locations” (Ex. 1001, 10:39–47) suggests that a resource tag buffer need not be one particular area of memory. During oral argument,

Patent Owner confirmed this and stated that the resource tag buffer “must be a unit” but “need not be a contiguous unit.” Tr. 32:13–14. Patent Owner further stated that the resource tag buffer “must, either logically or physically[,] be a unit.” Tr. 32:16–17. Thus, the resource tag buffer in the specification of the ’505 patent is made up of areas of memory that store the resource tags.

Patent Owner’s declarant, Dr. Przybylski, offers the following explanation of the term “buffer”:

The word “buffer” has several different uses in this and related arts. For example, a small circuit that amplifies or drives a single signal without inverting it is called a buffer too. In this specification, it is clear from the inventors’ use that they are using the term in its other principal meaning: a small to medium-sized structure for storing one or more multi-bit quantities, typically for a short or interim period of time. To be viewed as single buffer by a person of ordinary skill, the unit must be logically or physically a single entity. Logical unity involves storing one or more values of the same type, typically with a common input and common output. Alternatively, in the case a single buffer that holds a plurality of types of values, those values must be collected into a consistent higher-level structure that is replicated one or more times identically, and those sub-values must have a commonality of purpose or usage, again typically temporally and/or logically.

An example of such a structure is a Translation Look-aside Buffer, a common address translation block found in most processors. In its classical form, a TLB is a single unified physical structure that holds a plurality of identical address translation entries. Each entry includes a physical address, a virtual page component and some access control bits. So, although a TLB holds different types of quantities, they are organized logically and physically into a uniform array of identical structures. That uniformity is critical to a TLB as being considered as a single buffer instead of a plurality of separate buffers.

This is the same sort of structure that is disclosed in the '505 patent.

Ex. 2002 ¶¶ 38–40. Dr. Przybylski's position is that a translation look-aside buffer "is a single unified physical structure," which "is the same sort of structure that is disclosed in the '505 patent." Ex. 2002 ¶¶ 39–40. This testimony, however, conflicts with Patent Owner's acknowledgement that the resource tag buffer need not occupy contiguous memory locations in memory. See Tr. 32:9–17.

Dr. Przybylski also testifies that, "[t]o be viewed as single buffer by a person of ordinary skill, the unit must be logically or physically a single entity. Logical unity involves storing one or more values of the same type, typically with a common input and common output." Ex. 2002 ¶ 38. By this reasoning, all data of the same type comprise a buffer. Thus, the collection of resource tags stored in memory would comprise a "resource tag buffer." This is consistent with our statement in the Decision on Institution that "the specification and claim 2 of the '505 patent broadly describe the 'resource tag buffer' in terms of what it stores, namely resource tags." Dec. on Inst. 14. Importantly, Dr. Przybylski's "logical unity" definition does not impart any physical constraints on the buffer, reinforcing that no particular physical structure is required for the "resource tag buffer."

Having considered the intrinsic record as well as extrinsic evidence in Dr. Przybylski's testimony, we reiterate our statement in the Decision on Institution that "the specification and claim 2 of the '505 patent broadly describe the 'resource tag buffer' in terms of what it stores, namely resource tags." Dec. on Inst. 14. The '505 patent itself does not describe any particular physical structure for the buffer and, instead, discloses that it can be various locations in cache memory. Ex. 1001, 10:39–47. The defining

feature is the type of data stored, namely resource tags, and claims 2 and 15 recite the particular data that the resource tags store. Beyond this, it is difficult to define particularly “a resource tag buffer.” We, however, need not delineate the precise metes and bounds of “a resource tag buffer” to determine whether Petitioner has shown that the prior art’s disclosure is within the broadest reasonable interpretation of it. For the reasons explained below in the discussions of claims 2 and 15, we find each of Chauvel and Hughes teaches the claimed “resource tag buffer” even under Dr. Przybylski’s explanation of “buffer.”

2. Remaining Terms

Based on the record before us, we determine that the remaining terms of the claims at issue do not require express constructions at this time. *See, e.g., Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

C. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art;

(3) the level of ordinary skill in the art; and (4) any secondary considerations, if in evidence. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966).

*D. Obviousness over Chauvel
(Claims 1–18)*

Petitioner contends claims 1–18 of the ’505 patent are unpatentable under 35 U.S.C. § 103(a) as having been obvious over the teachings of Chauvel. Pet. 1, 10–45. Patent Owner provides arguments only with respect to the “resource tag buffer” limitations of dependent claims 2 and 15. PO Resp. 30–40; Tr. 16:25–17:4. Nevertheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware LLC, v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). For the reasons explained below, we are persuaded that the subject matter of claims 1–18 would have been obvious based on the teachings of Chauvel.

1. Chauvel

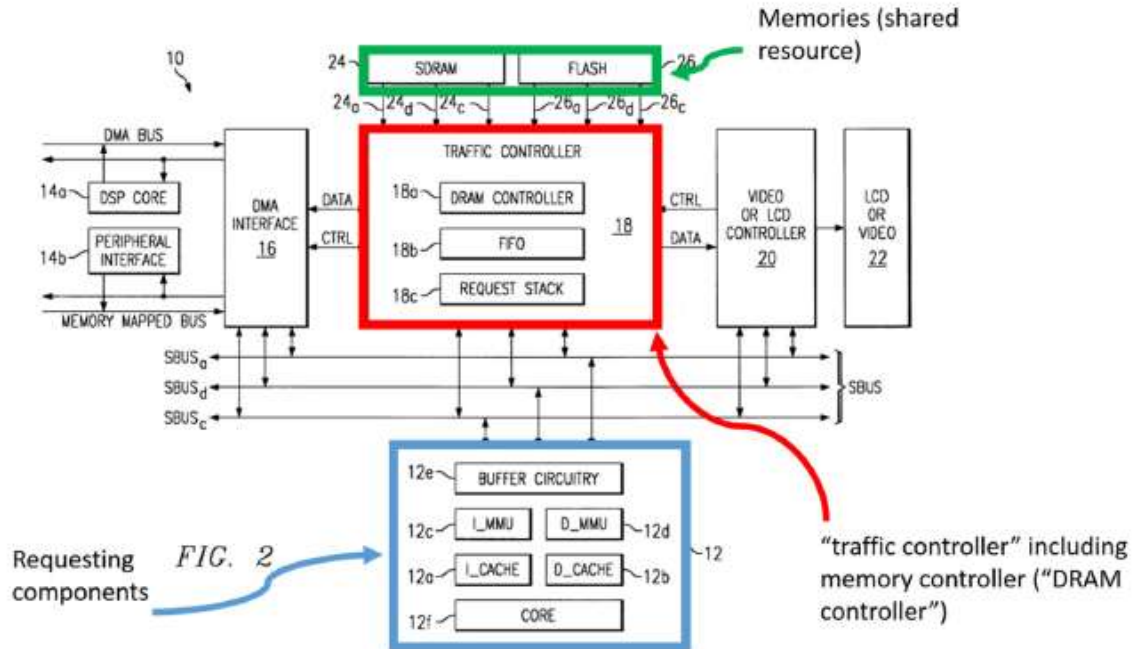
Chauvel “relate[s] to environments implementing memory control and direct memory access (‘DMA’), and are more particularly directed to circuits, systems, and methods in these environments for reducing access latency.” Ex. 1004, 1:28–32.

2. Independent Claims 1 and 14

a. “Universal resource access controller”

Independent claim 1 is reproduced above and is directed to “[a] universal resource access controller coupled to a requesting system and a resource.” Independent claim 14 recites, “In a universal resource access controller, a method of controlling access to a resource by a requesting system.”

With its obviousness analysis, Petitioner provides an annotated version of Chauvel's Figure 2, which is reproduced below.



Pet. 12. In this annotated version of Figure 2 from Chauvel, Petitioner identifies processor 12 as “Requesting components” coupled to traffic controller 18, which is coupled to SDRAM 24 and flash memory 26. *Id.* (citing Ex. 1004, 4:57–65, 5:5–11, 5:20–22, Fig. 2; Ex. 1002 ¶¶ 43–44).

Petitioner contends:

Although Chauvel’s specification generally describes controlling access to memory, “the various of the circuits and functional[ity] described . . . may be used in various other contexts.” [Ex. 1004,] 5:17–20. Accordingly, the controller provided by Chauvel describes a controller that capable of controlling any resource in a computer. [Ex. 1002] ¶ 45. Moreover, [a person of ordinary skill in the art] would find it obvious to modify the Chauvel controller to be universal in view of this teaching. *Id.* Specifically, [a person of ordinary skill in the art] would understand Chauvel’s teachings as generally applicable to control of resources beyond memory and would be motivated to use the improved technique of Chauvel throughout

the computer. *Id.* [A person of ordinary skill in the art] would recognize that doing so would improve the efficiency of the system by reducing the number of separately designed controllers, reducing the controller die size, connections, and enabling better and more efficient control and pipelining of resources. *See* [Ex. 1002] ¶ 45.

Id. at 12–13.

We are persuaded by Petitioner’s arguments, and we find, that Chauvel’s traffic controller 18 teaches a “universal access resource controller.” As we explained in our Decision on Institution, “a memory controller is within the broadest reasonable interpretation of a ‘universal resource access controller.’” Dec. on Inst. 5–6. Although the ’505 patent states that “the invention can be implemented as a universal controller capable of controlling access to any resource, shared or not” (Ex. 1001, 6:4–5), the ’505 patent describes a memory controller as “one possible embodiment of the universal controller 104” (Ex. 1001, 14:16–17). Throughout the ’505 patent, the disclosed subject matter is described in terms of a memory controller. *See generally* Ex. 1001.

Chauvel discloses that traffic controller 18’s “general operation along with various circuits coupled to it enable it to receive DMA access requests and direct access requests from host processor 12, and in response to both of those requests to permit transfers,” including transfers between host processor 12 (i.e., “requesting system”) and SDRAM 24 (i.e., “resource”) and transfers between host processor 12 (i.e., “requesting system”) and flash memory 26 (i.e., “resource”). Ex. 1004, 5:37–48. Thus, Chauvel teaches “[a] universal resource access controller coupled to a requesting system and a resource,” as recited in claim 1, and “a universal resource access

controller . . . controlling access to a resource by a requesting system,” as recited in claim 14.

Furthermore, we are persuaded by Petitioner’s contentions, supported by the testimony of Dr. Tredennick, that it would have been obvious to modify Chauvel’s traffic controller to control resources beyond memory. Dr. Tredennick testifies as follows:

[A] person of skill would understand that the teachings of Chauvel could be applied to control a broad array of computer components. Specifically, a person of ordinary skill in the art would understand from the “various other contexts” teaching of Chauvel that the aspects of control disclosed in Chauvel are generally applicable to the control of resources beyond memory. The benefits of doing so would be readily understood. Using a single universal controller improves the efficiency of the system by reducing the number of separately designed controllers, reduce the controller die size, connections, and better requirements, and enable better and more efficient control and pipelining of resources, as the universal controller would unify operations and control within the computer system. The need for a single “universal” controller with the efficiency taught by Chauvel is particularly needed in smaller and more portable devices, which have limited space for components.

Ex. 1002 ¶ 45. We credit this testimony, and we find that a person of ordinary skill in the art would have recognized the benefit of using Chauvel’s traffic controller to control resources in addition to memory and that a person of ordinary skill in the art would have been motivated to modify Chauvel’s traffic controller accordingly.

b. Generating a resource access request

Claim 1 further recites that, “when the requesting system desires access to the resource, the requesting system generates a resource access request which is passed to the universal resource controller.” Claim 14

similarly recites “generating a resource access request by the requesting system” and “passing the resource access request to the universal resource access controller.”

Petitioner contends Chauvel teaches this subject matter because it describes that the traffic controller receives memory access requests from various components including host processor 12 and, in response to those requests, permits data transfers from various components. Pet. 13 (citing Ex. 1004, 3:55–57, 5:37–48²; Ex. 1002 ¶ 46). We are persuaded by this contention. Chauvel discloses that

traffic controller 18[’s] . . . general operation along with various circuits coupled to it enable it to receive DMA access requests and direct access requests from host processor 12, and in response to both of those requests to permit transfers from/to the following:

host processor 12 from/to SDRAM 24

host processor 12 from/to flash memory 26.

Ex. 1004, 5:37–44. Thus, Chauvel describes that host processor 12 (i.e., “requesting system”) sends DMA access requests to traffic controller 18, and, therefore, we find Chauvel teaches “when the requesting system desires access to the resource, the requesting system generates a resource access request which is passed to the universal resource controller,” as recited in claim 1, and “generating a resource access request by the requesting system” and “passing the resource access request to the universal resource access controller,” as recited in claim 14.

² We understand Petitioner’s citation of “5:37–18” of Chauvel to be a typographical error. Instead, we understand Petitioner’s citation to be to column 5, lines 37–48 of Chauvel. See Ex. 1002 ¶ 46 (citing Ex. 1004, 5:37–48).

c. Generating a sequenced resource access command

Claim 1 still further recites that the “universal resource controller . . . in turn, uses a specific characteristic operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource to generate a corresponding sequenced command suitable for accessing the resource as required by the requesting system.” Claim 14 similarly recites “generating a sequenced resource access command suitable for accessing the resource as required by the requesting system and based upon a specific operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource.”

Petitioner contends Chauvel’s disclosure of the use of CAS latency for a memory command teaches using a “specific characteristic operating parameter of the requested resource.” Pet. 14–16. We are persuaded by this contention. For a read command, Chauvel discloses that “the read access command is aligned to be issued on the first cycle following the last data cycle of the current access CA, minus the CAS latency for the read” and that, “in most systems, it is contemplated that the CAS latency may be 1, 2, 3, or 4 cycles depending on the memory being accessed and clock frequency.” Ex. 1004, 10:10–16. Chauvel also discloses that “compare logic and state machine 30 [of DRAM controller 18a] includes an indicator of the current bus frequency, and from that frequency a corresponding CAS latency is selected.” *Id.* at 10:20–23. As noted by Petitioner, the ’505 patent identifies CAS latency as an example of an operating characteristic. Pet. 14 (quoting Ex. 1001, 8:56–60); *see* Ex. 1001, 8:56–60 (“[I]t is recognized that each shared resource has associated with it a set of operating characteristics

(such as access time, CAS latency in the case of DRAM type devices, for example).”). Based on this evidence, we find Chauvel’s disclosure of the use of CAS latency for a memory command teaches using “a specific characteristic operating parameter of the requested resource.”

Petitioner further contends Chauvel discloses using both a current state of the requested resource and a requested state of the requested resource to generate a memory command. Pet. 16–21. As for the “current state” of the resource, Petitioner cites Chauvel’s disclosure that DRAM controller 18a “includes circuitry sufficient to indicate various state information which identifies the current operation of SDRAM 24” and “includes a CURR_ACCESS register which stores information relating to the most recent (or current) request which has been given access to SDRAM 24.” Ex. 1004, 7:26–28, 7:62–65, *quoted in* Pet. 16–17. Chauvel states that “[t]his information includes the remaining part of the address of the current access (i.e., the column address, its direction, and size).” Ex. 1004, 7:65–67, *quoted in* Pet. 17. As for the “requested state” of the resource, Petitioner cites Chauvel’s disclosure that DRAM controller 18a “includes an input 28 for receiving a next (i.e., pending) access request” and that “[t]he access request information received at input 28 is presented to a compare logic and state machine 30, which also has access to . . . the information stored in the CURR_ACCESS register.” *Id.* at 7:67–8:8, *quoted in* Pet. 18–19.

Petitioner contends the operations depicted in the flowchart of Figure 4 show that both the current state and the requested state of the resource are used to generate a memory command. Pet. 18, 20–21. We are persuaded by Petitioner’s contentions that Chauvel teaches using the current

state and the requested state of the resource to generate a memory command.

Figure 4 of Chauvel is reproduced below.

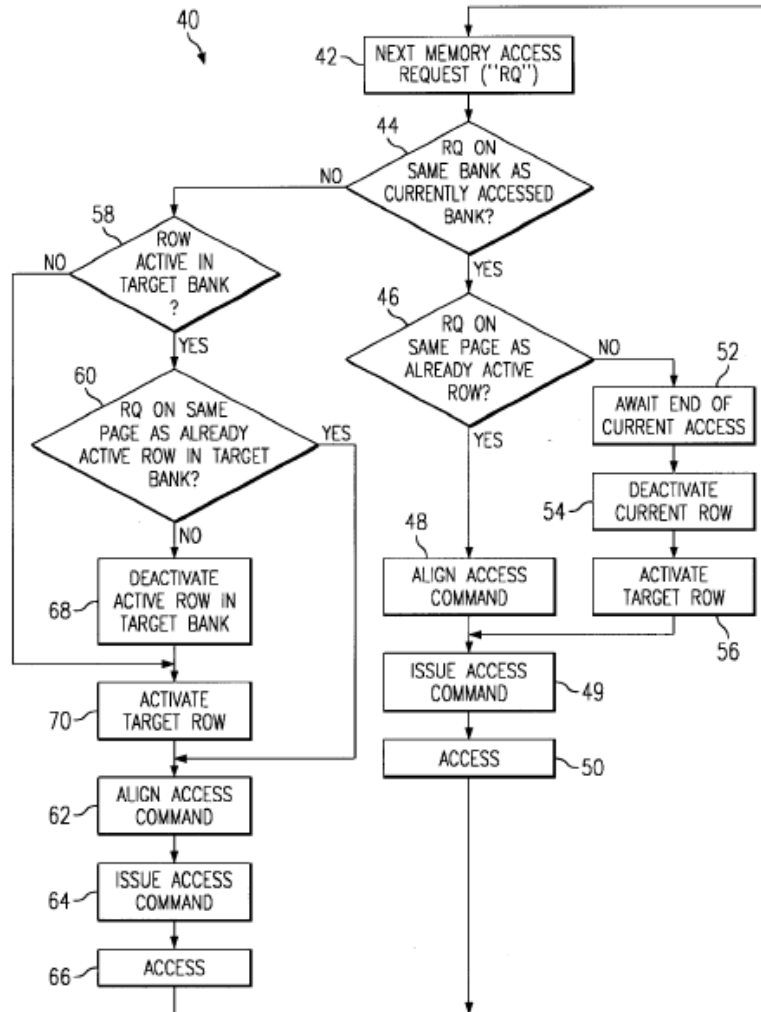


FIG. 4

Figure 4 “illustrates a flow chart of an embodiment of processing memory access requests by DRAM controller 18d to reduce system latency.”

Ex. 1004, 4:25–27.

Referring to Figure 4, Chauvel discloses that “[m]ethod 40 commences with a step 42 where the next memory access request (abbreviated ‘RQ’) is selected for analysis” and that “[s]tep 44 determines whether the bank to be accessed by the RQ from step 42 (hereafter referred

to as the target bank) is on the same bank as is currently being accessed.”
Ex. 1004, 8:54–56, 9:10–12. If the request is to the same memory bank, processing proceeds to step 46, which “determines whether the target page (or row) is on the same page (or row) as is already active in the target bank.”
Ex. 1004, 9:24–34. If the target row is already active, processing proceeds to step 48 (aligning an access command), step 49 (issuing an access command), and step 50 (performing access). *Id.* at Fig. 4, 9:41–43. By this procedure, Chauvel notes that “the operation of steps 48 and 50 is such that the active row is maintained active and for both the first and all consecutive accesses directed to the same row on the same memory bank” such that “there is no additional step of precharging the row between the occurrence of these accesses.” *Id.* at 10:60–64.

Chauvel, therefore, describes looking at the current state of the memory to see (1) if it is already active and (2) if it matches the requested state of the resource for the next operation, which requires an active state as well. Based on this disclosure, we find Chauvel teaches using “a current state of the requested resource” and “a requested state of the requested resource” for a memory command.

Petitioner further contends Chauvel teaches “generat[ing] a corresponding sequenced command suitable for accessing the resource as required by the requesting system.” Pet. 22–24. According to Petitioner, Chauvel describes a sequence of operations to perform a memory access that includes precharging a row, activating the row, and issuing an access command. *Id.* at 22 (citing Ex. 1004, 11:43–50; Ex. 1002 ¶ 62). Referring to memory accesses A1–A4 in Figure 6 of Chauvel, Petitioner argues

Chauvel teaches “set[ting] the timing between individual commands being sent to the memory.” *Id.* at 23.

We are persuaded by these contentions. As discussed above, we find Chauvel’s disclosure of CAS latency teaches “a specific characteristic operating parameter of the requested resource.” Chauvel describes that, for a read command in step 48 of Figure 4, “the read access command is aligned to be issued on the first cycle following the last data cycle of the current access CA, minus the CAS latency for the read.” Ex. 1004, 10:10–14. Accesses A1–A4 in Figure 6 include “overhead,” which Chauvel discloses “may arise from various control operations, such as precharging memory rows and/or banks as well as receipt of a signal indicating the end of an access.” Ex. 1004, 2:60–63, *quoted in* Pet. 24. Thus, Chauvel describes, in the process of Figure 4, using a specific operating parameter of the memory, such as CAS latency, and current and requested states of the memory to generate a command to the memory.

We find, therefore, that Chauvel teaches that the “universal resource controller . . . in turn, uses a specific characteristic operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource to generate a corresponding sequenced command suitable for accessing the resource as required by the requesting system,” as recited in claim 1, and “generating a sequenced resource access command suitable for accessing the resource as required by the requesting system and based upon a specific operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource,” as recited in claim 14.

d. Conclusion as to claims 1 and 14

In summary, we find Chauvel teaches all of the limitations of claims 1 and 14. Patent Owner does not present any objective evidence of nonobviousness as to any of the challenged claims. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 1 and 14 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

3. Claims 2 and 15

Claims 2 and 15 depend, respectively, from claims 1 and 14 and recite that the “universal resource access controller” comprises various elements, which we discuss below.

a. Configurable system interface

Claims 2 and 15 recite “a configurable system interface coupled to the requesting system suitably arranged to both receive the resource access request and to generate a corresponding universal command.” Petitioner contends, and we are persuaded, that Chauvel’s DRAM controller 18a teaches the recited interface. Pet. 25 (citing Ex. 1004, 5:12–24, 8:2–4, 8:12–16, Fig. 3; Ex. 1002 ¶ 70). Chauvel discloses that “traffic controller 18 includes a DRAM controller 18a as an interface for the connection between traffic controller 18 and SDRAM 24.” Ex. 1004, 5:12–14. Figure 3 provides a block diagram of the DRAM controller and shows “access request information received at input 28,” thus teaching receiving a request. Ex. 1004, 8:2–3. Chauvel further discloses that “compare logic and state machine 30 [of DRAM controller 18a] is connected to provide an address to address bus 24_a between DRAM controller 18a and SDRAM 24, and to provide control signals to control bus 24_c between DRAM controller 18a and

SDRAM 24.” Ex. 1004, 8:12–16. Thus, Chauvel teaches that DRAM controller generates a command for the DRAM and sends it to the DRAM.

Based on this evidence, therefore, we find Chauvel teaches “a configurable system interface coupled to the requesting system suitably arranged to both receive the resource access request and to generate a corresponding universal command.”

b. Universal command sequencer

Claims 2 and 15 further recite “a universal command sequencer coupled to the configurable system interface.” Petitioner contends, and we are persuaded, that Chauvel’s state machine 30 generates the sequenced command and, therefore, teaches this limitation. Pet. 26 (citing Ex. 1004, 8:2–4, 8:49–55, Fig. 3; Ex. 1002 ¶ 72). In particular, Chauvel discloses that the method of Figure 4 showing how memory requests are processed “is accomplished through the operation generally of compare logic and state machine 30.” Ex. 1004, 8:49–54.

Based on this evidence, therefore, we find Chauvel teaches “a universal command sequencer coupled to the configurable system interface.”

c. Resource tag buffer

Claims 2 and 15 further recite “a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current state of the requested resource and a resource tag arranged to identify the requested state of the requested resource.”

Petitioner contends that, “[a]ccording to the specification of the ’505 patent, the ‘resource tag buffer’ stores information about the current state and the requested state of the shared resource.” Pet. 26 (citing Ex. 1001, 7:3–29, 7:46–60). Petitioner contends Chauvel discloses that

traffic controller 18 stores information about the current state of the resource, most recent access, and pending access requests. *Id.* at 27–28. In particular, Petitioner argues Chauvel discloses that page or row registers³ “stor[e] information about the current state of the requested resource (i.e. which pages in which banks are currently active),” that “a ‘current access register’ . . . stor[es] information about the most-recent access,” and that request stack 18c stores pending requests. *Id.*

Patent Owner argues that the multiple different storage structures identified by Petitioner as storing the resource tags cannot be considered “a resource tag buffer.” PO Resp. 36–40. Although Chauvel depicts the identified storage areas in different parts of traffic controller 18, we find that they comprise “a resource tag buffer” as claimed. As discussed above in the section addressing claim construction, the ’505 patent does not describe any particular structure for the resource tag buffer but, rather, describes the buffer in terms of what it stores, namely resource tags. *See* Section II.B.1 above. Patent Owner does not dispute that the identified structures store the claimed resource tags. As discussed with respect to claim 1, Chauvel describes looking at the current state of the memory to see if it is already active and if it matches the requested state of the resource for the next operation, i.e., an active state. *See* Section II.D.2.c above; *see also* Ex. 1004, 8:54–56, 9:10–12, 9:24–34, 9:41–43, 10:60–64, Fig. 4. Thus, Chauvel teaches that the current and requested states of the requested resource are

³ Although Petitioner refers to “page registers” in Chauvel (Pet. 27), Patent Owner correctly notes that Chauvel actually “refer[s] to ‘row registers’ ([Ex. 1004,]7:20) and equates rows and pages (*see id.* at 9:29-30)” (PO Resp. 32).

stored in traffic controller 18. The '505 patent itself describes resource tags being “located in disparate locations” (Ex. 1001, 10:39–47), and Patent Owner’s counsel confirmed that resource tag buffer “need not be a contiguous unit” (Tr. 32:13–14). We find that Chauvel’s storage locations for the current and requested states of the requested resource teach “a resource tag buffer.”

Furthermore, Chauvel teaches “a resource tag buffer” under Dr. Przybylski’s definition of “buffer.” In particular, Dr. Przybylski testifies, “To be viewed as single buffer by a person of ordinary skill, the unit must be logically or physically a single entity. Logical unity involves storing one or more values of the same type, typically with a common input and common output.” Ex. 2002 ¶ 38; *see also* Tr. 32:16–17 (Patent Owner’s counsel stating that a buffer “must, either logically or physically[,] be a unit”). Chauvel describes such “logical unity.” For example, Chauvel discloses that DRAM controller 18a “includes an input 28 for receiving a next (i.e., pending) access request” and that “[t]he access request information received at input 28 is presented to a compare logic and state machine 30, which also has access to . . . the information stored in the CURR_ACCESS register.” Ex. 1004, 7:67–8:8, *quoted in* Pet. 18–19. As discussed with respect to claim 1, Chauvel discloses determining whether the memory request is for an already active memory row. Ex. 1004, 8:54–56, 9:10–12, 9:24–34, 9:41–43, Fig. 4 (step 46). Chauvel discloses the following:

Step 44 determines whether the bank to be accessed by the RQ from step 42 (hereafter referred to as the target bank) is on the same bank as is currently being accessed. Compare logic and state machine 30 makes this determination by comparing the

bank portion of the address in the RQ with the bank portion of the address stored in the CURR_ACCESS register. . . .

Step 46 determines, with it now found that the target bank of the RQ is on the same bank as the bank currently being accessed, whether the page to be accessed by the RQ (hereafter referred to as the target page) is on the same row as is already active in the target bank. . . . Compare logic and state machine 30 makes this determination by comparing the page address portion of the address in the RQ with the corresponding bits in the active row address stored in the appropriate register for the target bank.

Ex. 1004, 9:10–38. Thus, the values of the requested state and the current state are of the same type such that Chauvel’s compare logic and state machine 30 can compare them. See Ex. 2002 ¶ 38 (“Logical unity involves storing one or more values of the same type”). We, therefore, agree with Petitioner that, “even under Patent Owner’s construction, the storage structures which store information about the current and requested state of the resource are logically related because they both contain the same type of information about the resource—namely the state of the resource.” Pet. Reply 22.

Based on the foregoing, we find Chauvel teaches “a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current state of the requested resource and a resource tag arranged to identify the requested state of the requested resource.”

d. Characteristic operating parameter buffer

Claims 2 and 15 further recite “a characteristic operating parameter buffer coupled to the command sequencer arranged to store the characteristic operating parameter associated with the requested resource.” Petitioner contends, and we are persuaded, that Chauvel teaches this subject matter by

its disclosure of storing CAS latency. Pet. 28–29 (citing Ex. 1004, 10:20–24, 13:51–56, 14:55–61, claim 11; Ex. 1002 ¶ 77). Chauvel discloses that “compare logic and state machine 30 includes an indicator of the current bus frequency, and from that frequency a corresponding CAS latency is selected.” Ex. 1004, 10:20–23. Chauvel further discloses that latency information “correspond[s] to states in compare logic and state machine 30, and those states are encoded for each row in the latency signal LAT_Rn.” Ex. 1004, 14:55–59. We find that this disclosure of latency information corresponding to states stored in logic and state machine 30 teaches “a characteristic operating parameter buffer coupled to the command sequencer arranged to store the characteristic operating parameter associated with the requested resource.”

e. Generating the sequenced universal command

Claim 2 further recites, “wherein the universal command sequencer uses the respective resource tags that identify the current state and the requested state of the requested resource and the characteristic operating parameter associated with the requested resource to generate the sequenced universal command.” Petitioner contends Chauvel teaches this subject matter for the same reasons that it teaches generating a sequenced command using this information, as set forth with respect to claim 1. Pet. 29. We agree, and we find Chauvel teaches this subject matter for the reasons discussed with respect to claim 1. See Section II.D.2.c above.

f. Conclusion as to claims 2 and 15

In summary, we find Chauvel teaches all of the limitations of claims 2 and 15. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the

subject matter of claims 2 and 15 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

4. *Claims 3 and 16*

Claim 3 and 16, which depend from claims 2 and 14, respectively, recite, “wherein the requesting system is one of a plurality of processors included in a multiprocessor computing system, wherein the configurable system interface is configurable to be able to accept and process resource requests from any of the plurality of processors.” Citing Chauvel’s disclosure of general purpose processor 12 and digital signal processor (“DSP”) 14a, Petitioner contends “Chauvel discloses or renders obvious a multiprocessor system in which the requesting system is one of a plurality of processors.” Pet. 29–30 (citing Ex. 1004, 4:57–67,⁴ 5:24–35, Fig. 2; Ex. 1002 ¶¶ 79–80). Petitioner further contends “[t]he processor(s) in Chauvel is capable of issuing a processor command to the system bus, which is transmitted to the traffic controller.” *Id.* at 39.

We are persuaded by these contentions because Chauvel discloses that “traffic controller 18” may “present access requests to SDRAM 24 in response to access requests posed to traffic controller 18 from the various circuits which may request DMA access or direct access (e.g., host processor 12, DSP 14a[]).” Ex. 1004, 22:28–33. Thus, we find Chauvel teaches a system in which multiple processors, namely host processor 12 and DSP 14a, may request access via the traffic controller. Furthermore, as discussed with respect to claim 2, we find Chauvel’s DRAM controller 18a teaches the “configurable system interface.” We further find that the DRAM

⁴ Although Petitioner cites column 2, based on our review of the reference, it appears that Petitioner intended to cite column 4.

controller accepts and processes the requests because, as discussed with respect to claim 2, Chauvel teaches that DRAM controller generates a command for the DRAM and sends it to the DRAM. Ex. 1004, 8:12–16.

Thus, we find Chauvel teaches the subject matter of claims 3 and 16. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 3 and 16 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

5. *Claim 4*

Claim 4 recites,

A universal resource access controller as recited in claim 3, wherein the requested resource is one of a plurality of shared resources wherein each of the plurality of shared resources has associated with it a characteristic operating parameter which is stored in a corresponding operating characteristic parameter buffer and wherein each of the plurality of shared resources has associated with it a resource tag indicating a current state of each of the plurality of resources and a resource tag indicating a requested state of each of the plurality of resources.

Referring to portions of its contentions for claims 1 and 2, Petitioner contends this subject matter would have been obvious based on Chauvel’s disclosure of traffic controller 18’s interfacing with both SDRAM 24 and flash memory 26. Pet. 31–33 (citing Ex. 1004, 6:27–39, Fig. 2; Ex. 1002 ¶¶ 84–85). In particular, Petitioner cites Chauvel’s disclosure that “traffic controller 18 includes circuits to support the connections to the various circuits described above,” including “a flash memory interface which generates the appropriate signals required by flash devices” and “DRAM controller 18a . . . which implements the control of a state machine and generates the appropriate signals required by SDRAM 24.” Ex. 1004, 6:27–

35, *cited in* Pet. 32. Thus, Chauvel teaches that current and requested states of these memories are known for purposes of controlling the memory and generating such requests. Ex. 1004, 6:27–35. Petitioner also contends “Chauvel teaches that it[] stores . . . the appropriate timing characteristics for the particular memory being used.” Pet. 32 (citing Pet. 14–16, 28–29; Ex. 1002 ¶ 84). We agree because, as discussed above for claim 1, Chauvel discloses using CAS latency in generating a memory command, which teaches using “a specific characteristic operating parameter of the requested resource.” Ex. 1004, 10:10–16, 10:20–23.

Thus, we find Chauvel teaches the subject matter of claim 4. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 4 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

6. *Claim 5*

Claim 5 recites,

A universal resource access controller as recited in claim 4, further comprising:

an address space controller coupled to the command sequencer arranged to store each of the operating characteristic parameter buffers associated with each of the plurality of shared resources; and wherein the resource tag buffer stores the resource tags associated with each of the plurality of shared resources.

Petitioner argues Chauvel’s disclosure of circuitry in Figure 7 that stores latency signals (i.e., operating characteristic parameters) teaches “an address space controller” as claimed. Pet. 33–35 (citing Ex. 1004, 13:59–66, 14:14–32, Fig. 7; Ex. 1002 ¶¶ 87–90). As to the “wherein” clause, Petitioner refers to its contentions with respect to claims 2 and 4. Pet. 32–33, 35.

We are persuaded by Petitioner’s contentions. Referring to Figure 7, Chauvel describes circuitry for processing an address of an access request. Ex. 1004, 13:59–61. Figure 7 depicts latency signals contained in registers. *Id.* at 14:44–47. As discussed with respect to claim 2, we find Chauvel’s disclosure of latency information teaches “a characteristic operating parameter buffer.” As discussed for claims 2 and 4, we find Chauvel teaches resource tags having current and requested states of memories that are stored in a resource tag buffer.

Thus, we find Chauvel teaches the subject matter of claim 5. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 5 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

7. *Claim 6*

Claim 6 recites,

A universal resource access controller as recited in claim 5 arranged in such a way that when a particular requesting system desires access to a particular one of the plurality of shared resources, the requesting system is identified to the universal resource access controller which, in response, configures the configurable system interface to receive a particular shared resource request generated by the requesting system and in turn to generate a corresponding universal command, and wherein the command sequencer generates a sequenced universal command specific to the requested shared resource based upon the current state of the requested resource and the requested state of the requested resource as indicated by the corresponding resource tags and the operating characteristic parameter associated with the requested resource.

Citing Chauvel’s disclosure of receiving requests having priorities based on the requesting circuit, Petitioner contends “Chauvel teaches that the

SDRAM controller is configured to receive high priority requests from particular devices.” Pet. 35–36 (citing Ex. 1004, 15:7–10, 15:19–25, 15:37–40, Table 2; Ex. 1002 ¶¶ 92–94). Chauvel discloses that “[t]raffic controller 18 also includes a priority handler and state machine 18d,” which includes “a priority table” that “lists the order in which access requests are serviced by issuing corresponding requests to DRAM controller 18a.” Ex. 1004, 15:28–37. Therefore, we find Chauvel teaches configuring the interface to handle requests of varying priority as such requests are received.

As to the recitation of “generat[ing] a sequenced universal command,” Petitioner refers to its analysis of similar subject matter in claim 1. Pet. 36. For the reasons discussed with respect to claim 1, we find Chauvel teaches this subject matter.

Thus, we find Chauvel teaches the subject matter of claim 6. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 6 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

8. Claim 7

Claim 7 recites, “A universal resource access controller as recited in claim 6, wherein the requesting system is a processor arranged to perform executable instructions.” As discussed for claim 3, we find Chauvel’s disclosure of host processor 12 requesting access to memory teaches that the requesting system is a processor. We agree with Petitioner that host processor 12 is “arranged to perform executable instructions” because Chauvel discloses that host processor 12 has “instruction cache 12a” (Ex. 1004, 4:58–59). Pet. 37 (citing Ex. 1004, 4:57–63; Ex. 1002 ¶ 96).

Thus, we find Chauvel teaches the subject matter of claim 7. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 7 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

9. Claims 8 and 9

Claim 8 recites, “A universal resource access controller as recited in claim 7, wherein the shared resource is a peripheral bus used to interconnect a plurality of computing system peripheral devices.” Claim 9 recites,

A universal resource access controller as recited in claim 8, wherein when the processor desires to access the peripheral bus, the processor generates a peripheral bus request that is converted by the universal resource access controller into a corresponding peripheral bus access request that is based upon the current state and the requested state of the peripheral bus and the operating characteristics of the peripheral bus.

Citing the testimony of Dr. Tredennick, Petitioner contends this subject matter would have been obvious to a person of ordinary skill in the art. Pet. 37–38 (citing Ex. 1002 ¶¶ 97–100; Ex. 1004, 5:17–20). Dr. Tredennick cites Chauvel’s disclosure of peripheral interface 14b in Figure 2. Ex. 1002 ¶¶ 98–99 (citing Ex. 1004, 4:64–5:1, 5:3–4, 5:46–48, Fig. 2). Chauvel discloses, “Although not shown, various peripheral devices may therefore be coupled to peripheral interface 14b.” Ex. 1004, 4:67–5:1. Based on this disclosure of Chauvel and the testimony of Dr. Tredennick, we are persuaded that it would have been obvious to a person of ordinary skill in the art that “the shared resource is a peripheral bus used to interconnect a plurality of computing system peripheral devices.” We also are persuaded that generating a peripheral bus access request, as recited in claim 9, would

have been obvious to a person of ordinary skill in the art based on Chauvel's disclosure of generating a resource access request for a requested resource, as discussed for claim 1.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 8 and 9 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

10. Claim 10

Claim 10 recites, "A universal resource access controller as recited in claim 9, wherein the processor is one of a plurality of processors included in multiprocessor computing system." Petitioner refers to its analysis for claim 3. Pet. 38. As discussed for claim 3, we find Chauvel teaches a system in which multiple processors, namely host processor 12 and DSP 14a, may request access via the traffic controller. Therefore, each of these processors is "is one of a plurality of processors included in multiprocessor computing system."

Thus, we find Chauvel teaches the subject matter of claim 10. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 10 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

11. Claim 11

Claim 11 is directed to "[a] multi-processor computer system" having "a system bus" and "a plurality of processors each capable of issuing a processor command and an associated data burst to the system bus." Petitioner contends Chauvel's SBUS teaches a "system bus." Pet. 38–39

(citing Ex. 1004, Fig. 2). We agree because Chauvel states that “SBUS” is “a system bus.” Ex. 1004, 4:62–63. As discussed for claim 3, we find Chauvel teaches a system having a plurality of processors. Petitioner contends Chauvel teaches processors that are capable of issuing a processor command to the SBUS and also teaches data bursts. Pet. 39 (citing Ex. 1004, 19:55–20:18, 27:7–28:9, claims 23 and 24; Ex. 1002 ¶ 105). We are persuaded this subject matter would have been obvious to a person of ordinary skill in the art based on Chauvel’s disclosures of multiple processors, communication among components of the system via the SBUS, and handling of burst requests. Ex. 1004, 4:57–5:7, 19:55–20:18, Fig. 2.

Claim 11 further recites, “a memory controller as recited in claim 1⁵ connected to the plurality of processors by way of the system bus, the memory controller being arranged to receive the processor commands and associated data bursts issued by the processors and to issue corresponding shared memory commands,” and “a shared memory device.” Petitioner contends Chauvel’s traffic controller receives processor commands, and we agree for the reasons discussed for claim 1. We also agree Chauvel discloses that its traffic controller handles burst requests. Ex. 1004, 19:55–20:18; *see id.* at 19:56–58 (“FIG. 10 illustrates a method 90 also performed by priority handler and state machine 18d, and directed to burst requests.”). For the reasons discussed as to claim 1, we find Chauvel discloses that its traffic controller issues commands to shared memory.

Claim 11 further recites “a command bus connecting the shared memory device to the memory controller arranged to carry the issued shared

⁵ Claim 1 recites a “universal resource access controller,” not a memory controller. We understand claim 11 to refer to the controller of claim 1.

memory commands according to a minimum issue time generated by the memory controller” and “a data bus connecting the shared memory to the memory controller arranged to carry read data from the shared memory to the memory controller based upon the data offset generated by the memory controller.” Petitioner cites Chauvel’s control bus 24_c and data bus 24_d, respectively, for teaching the claimed “command bus” and “data bus.” Pet. 40–44 (citing Ex. 1004, 5:5–11, 9:51–53, 14:48–55, Fig. 2; Ex. 1002 ¶¶ 108–112). We agree Chauvel teaches a “command bus” and a “data bus” because Chauvel discloses that “[t]raffic controller 18 is coupled via address 24_a, data 24_d, and control 24_c buses to a main memory which in the preferred embodiment is a synchronous dynamic random access memory (‘SDRAM’) 24.” Ex. 1004, 5:8–11. Petitioner contends Chauvel discloses taking latency into account when issuing commands, such that control bus 24_c is “arranged to carry the issued shared memory commands according to a minimum issue time generated by the memory controller.” Pet. 41–42 (citing Ex. 1004, 14:48–55; Ex. 1002 ¶ 109). We are persuaded by this contention because Chauvel discloses “the state machine of compare logic and state machine preferably includes sufficient states to accommodate the latency requirements which arise due to the various different combinations of commands which may be issued to SDRAM 24.” Ex. 1004, 14:48–52. Petitioner also contends Chauvel discloses aligning the next command according to latency considerations including waiting for the previous data transfer cycle to complete, such that data bus 24_d is “arranged to carry read data from the shared memory to the memory controller based upon the data offset generated by the memory controller.” Pet. 43 (citing Ex. 1004, 9:51–53). We are persuaded by this contention because Chauvel discloses

“align[ing] the access command (e.g., READ or WRITE) for the RQ to occur during or near the final data transfer cycle of the current access.”

Ex. 1004, 9:51–53.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 11 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

12. Claims 12 and 17

Claim 12 recites, “The multi-processor computer system as recited in claim 11, wherein the shared memory is an SDRAM.” Claim 17 recites, “A method as recited in claim 14, wherein the requested resource is a memory device.” As discussed for claim 11, Chauvel discloses that SDRAM 24 is a shared memory accessed via traffic controller 18. Ex. 1004, 5:12–14 (“[T]raffic controller 18 includes a DRAM controller 18a as an interface for the connection between traffic controller 18 and SDRAM 24.”).

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 12 and 17 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

13. Claim 13

Claim 13 recites, “The multi-processor computer system as recited in claim 11, wherein the shared memory is a SLDRAM module having a plurality of SDRAMs.” Petitioner argues, and Dr. Tredennick testifies, that SLDRAM (synchronous link dynamic access memory) was a proposed enhanced version of SDRAM that was designed to be fully backward compatible but was never implemented. Pet. 44 (citing Ex. 1002 ¶ 114;

Ex. 1009); Ex. 1002 ¶¶ 114–115. Because of its backward compatibility, Dr. Tredennick testifies that “a person of skill could easily adapt Chauvel to use this improved version of SDRAM.” Ex. 1002 ¶ 115.

We find that SLDRAM’s improvements over SDRAM would have provided a reason for a person of ordinary skill in the art to use SLDRAM, and its backward compatibility would place such a modification well within the skill level of a person of ordinary skill in the art, as Dr. Tredennick testifies. Ex. 1002 ¶¶ 114–115. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 13 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

14. Claim 18

Claim 18 recites, “A method as recited in claim 17, wherein the memory device is shared amongst a number of memory devices each of which are coupled to the universal resource access controller.” As discussed for claim 4, Chauvel discloses that “traffic controller 18 includes circuits to support the connections to the various circuits described above,” including “a flash memory interface which generates the appropriate signals required by flash devices” and “DRAM controller 18a . . . which implements the control of a state machine and generates the appropriate signals required by SDRAM 24.” Ex. 1004, 6:27–35. Chauvel’s Figure 2 shows SDRAM 24 and flash 26 coupled to traffic controller 18. Thus, we find SDRAM 24 “is shared amongst a number of memory devices each of which are coupled to the universal resource access controller,” as recited in claim 18.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject

matter of claim 18 would have been obvious based on Chauvel under 35 U.S.C. § 103(a).

*E. Obviousness over Chauvel and Ramanujan
(Claims 3–13, 16, and 18)*

Petitioner contends claims 3–13, 16, and 18 of the '505 patent are unpatentable under 35 U.S.C. § 103(a) as having been obvious over the combined teachings of Chauvel and Ramanujan. Pet. 1, 45–50. Although Patent Owner does not provide separate arguments addressing this ground, the burden remains on Petitioner to demonstrate unpatentability. See *Dynamic Drinkware*, 800 F.3d at 1378.

Petitioner cites Ramanujan for additional teachings of the use of multiple processors and multiple memories. Pet. 45–50. Petitioner argues “it would have been obvious to modify Chauvel based on Ramanujan to contain multiple central processing units” and “to contain multiple RAM memory modules.” *Id.* at 46. Petitioner contends “the benefits of Chauvel’s approach would be useful when applied within a multiprocessors system environment, as disclosed by Ramanujan, where [a person of ordinary skill in the art] would expect to achieve increased bandwidth and reduced latency of the memory system.” *Id.* at 47 (citing Ex. 1002 ¶ 132). Petitioner further contends “[i]mplementing the invention of Chauvel in the context of a multi-processor and multi-memory environment would have been trivial for [a person of ordinary skill in the art], and would involve only minor modifications,” and Petitioner explains how Chauvel’s system would have been modified in view of Ramanujan’s teachings regarding multiple processors and memories. *Id.* at 48–50 (citing Ex. 1002 ¶¶ 128, 130, 131).

As discussed above, we conclude the subject matter of claims 1–18 would have been obvious based on the teachings of Chauvel. See Section

II.D above. We find a person of ordinary skill in the art would have been motivated to combine Ramanujan's teachings of systems with multiple processors and multiple memories with Chauvel's teachings. In particular, we credit Dr. Tredennick's testimony as to why a person of ordinary skill in the art would have combined the teachings of Chauvel and Ramanujan. Ex. 1002 ¶¶ 127, 130, 132. For example, Dr. Tredennick testifies that "it was well known at the time that using multiple lower-capacity, but lower-cost memory modules instead of a single larger-capacity memory module could significantly reduce the cost of a computer." Ex. 1002 ¶ 130. Dr. Tredennick also testifies that, although Chauvel discloses one general purpose processor in its preferred embodiment, Chauvel states that its disclosure can be used in other contexts, and Dr. Tredennick explains that computer systems having multiple processors were well known at the relevant time, as taught by Ramanujan. Ex. 1002 ¶¶ 125–127 (citing Ex. 1004, 4:52–57, 5:14–20; Ex. 1005, Fig. 2).

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 3–13, 16, and 18 would have been obvious based on the combined teachings of Chauvel and Ramanujan under 35 U.S.C. § 103(a).

*F. Obviousness over Paluch and Hughes
(Claims 1–7 and 11–18)*

Petitioner also contends claims 1–7 and 11–18 of the '505 patent are unpatentable under 35 U.S.C. § 103(a) as having been obvious over the combined teachings of Paluch and Hughes. Pet. 1–2, 51–87. Patent Owner provides arguments only with respect to the "resource tag buffer" limitations of dependent claims 2 and 15. PO Resp. 40–46; Tr. 16:25–17:4.

Nevertheless, the burden remains on Petitioner to demonstrate

unpatentability. *See Dynamic Drinkware*, 800 F.3d at 1378. For the reasons explained below, we are persuaded that the subject matter of claims 1–4 and 11–18 would have been obvious based on the combined teachings of Paluch and Hughes.

1. *Paluch*

Paluch “relates to a memory controller for controlling and interfacing digital memory used in digital video processing.” Ex. 1006, 1:5–7. Paluch discloses, “To achieve a speed enhancement, the memory controller arbitrates requests for access to the memory such that, if possible, sequential memory accesses are directed to alternating memory banks.” Ex. 1006, 1:58–61.

2. *Hughes*

Hughes “relates to memory systems in data processors and more particularly relates to high performance data processing systems, such as network intermediate systems, having shared memory.” Ex. 1008, 1:10–13. Hughes discloses:

The present invention provides a shared memory architecture based on the use of synchronous dynamic random access memory SDRAM as the shared memory. The SDRAM based shared memory executes read, write and refresh requests for access to the memory in a pipeline fashion. In the shared memory architecture, a plurality of data paths are coupled to the shared memory which generate requests for access to the shared memory. These requests have characteristics including a starting address, a length and an access type, which are processed in a memory controller coupled to the shared memory. The memory controller stores requests from the plurality of paths, supplies a current request for access to the shared memory pipeline, and selects a next request from the stored request for access to the shared memory pipeline. The memory controller includes logic responsive to the characteristics of the request or requests

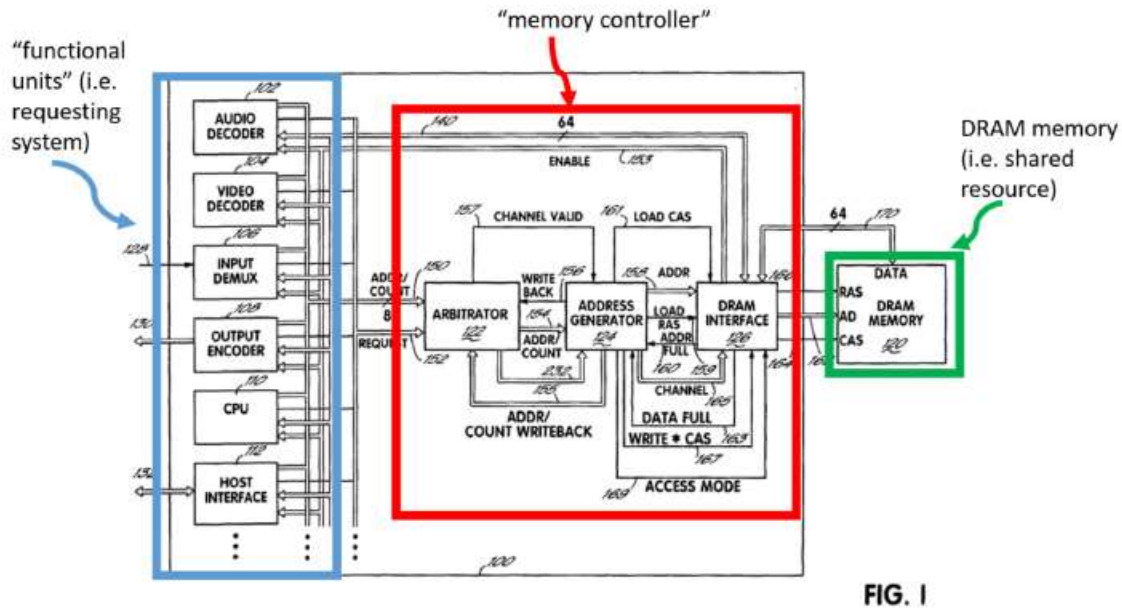
currently being executed in the shared memory pipeline and to the characteristics of pending requests, to select a next request for access to improve pipeline fullness, according to one embodiment of the invention.

Ex. 1008, 2:10–31.

3. Independent Claims 1 and 14

a. “Universal resource access controller”

As discussed above, independent claim 1 is directed to “[a] universal resource access controller coupled to a requesting system and a resource,” and independent claim 14 recites, “In a universal resource access controller, a method of controlling access to a resource by a requesting system.” With its obviousness analysis, Petitioner provides an annotated version of Paluch’s Figure 1, which is reproduced below.



Pet. 55. In this annotated version of Figure 1 from Paluch, Petitioner identifies units 102–112 as a “requesting system” coupled to a memory controller, which is coupled to DRAM 120. *Id.* (citing Ex. 1006, 3:27–31, 3:36–37, 4:6–11, Fig. 1; Ex. 1002 ¶¶ 143–144).

We are persuaded, and we find, that Paluch’s memory controller teaches a “universal access resource controller.” As we explained in our Decision on Institution, “a memory controller is within the broadest reasonable interpretation of a ‘universal resource access controller.’” Dec. on Inst. 5–6. Although the ’505 patent states that “the invention can be implemented as a universal controller capable of controlling access to any resource, shared or not” (Ex. 1001, 6:4–5), the ’505 patent describes a memory controller as “one possible embodiment of the universal controller 104” (Ex. 1001, 14:16–17). Throughout the ’505 patent, the disclosed subject matter is described in terms of a memory controller. *See generally* Ex. 1001.

Paluch discloses the following:

Access to DRAM 120 is controlled by a memory controller on the ASIC [(application specific integrated circuit)] 100. This memory controller includes an arbitrator 122 for determining which of the functional units 102 through 112 may access memory 120, an address generator 124 for generating addresses in response to requests for memory access from functional units 102 through 112, and a DRAM interface 126 for generating address and control signals for controlling DRAM 120.

Ex. 1006, 3:38–45. Thus, Petitioner’s identification of the memory controller as including arbitrator 122, address generator 124, and DRAM interface 126 in its annotated Figure 1 is consistent with Paluch’s description of the memory controller on ASIC 100. Furthermore, this passage shows that the functional units request access to the DRAM.

Therefore, Paluch teaches “[a] universal resource access controller coupled to a requesting system and a resource,” as recited in claim 1, and “a

universal resource access controller . . . controlling access to a resource by a requesting system,” as recited in claim 14.

b. Generating a resource access request

Claim 1 further recites that, “when the requesting system desires access to the resource, the requesting system generates a resource access request which is passed to the universal resource controller.” Claim 14 similarly recites “generating a resource access request by the requesting system” and “passing the resource access request to the universal resource access controller.”

Petitioner contends Paluch teaches this limitation because it describes that units 102–112 send memory access requests to the memory controller. Pet. 57 (citing Ex. 1006, 4:12–19, 6:4–7, 6:13–15; Ex. 1002 ¶ 148). We agree with Petitioner. Paluch discloses that “each functional unit 102 through 112 delivers virtual address and additional memory access request information through a bus 150” and that “each functional unit 102 through 112 requests service from the memory controller via signals on lines 152 connected to arbitrator 122.” Ex. 1006, 4:12–19.

Therefore, we find Paluch teaches “when the requesting system desires access to the resource, the requesting system generates a resource access request which is passed to the universal resource controller,” as recited in claim 1, and “generating a resource access request by the requesting system” and “passing the resource access request to the universal resource access controller,” as recited in claim 14.

c. Generating a sequenced resource access command

Claim 1 further recites that the “universal resource controller . . . in turn, uses a specific characteristic operating parameter of the requested

resource, a current state of the requested resource, and a requested state of the requested resource to generate a corresponding sequenced command suitable for accessing the resource as required by the requesting system.” Claim 14 similarly recites “generating a sequenced resource access command suitable for accessing the resource as required by the requesting system and based upon a specific operating parameter of the requested resource, a current state of the requested resource, and a requested state of the requested resource.”

Petitioner contends Paluch’s disclosure of the use of the memory’s timing specifications teaches using a “specific characteristic operating parameter of the requested resource.” Pet. 58–60 (citing Ex. 1006, 5:41–47, 5:60–6:2, 31:26–32, 33:65–67, Figs. 4C, 4D; Ex. 1002 ¶¶ 150–151, 153–154; Ex. 1011, 21). We are persuaded by this contention because Paluch discloses that “DRAM interface 126 further includes circuitry for producing column address strobe and row address strobe signals on lines 164 and 166 which are timed appropriately to optimize use of the DRAM 120.” Ex. 1006, 5:64–6:1.

Petitioner further contends Paluch teaches using both a current state of the requested resource and a requested state of the requested resource to generate a memory command. Pet. 61–63, 64–65. As for the “current state” of the resource, Petitioner cites Paluch’s disclosure that the memory controller “endeavors to alternate accesses between the banks of DRAM 120 when SDRAM memory is in use.” Ex. 1006, 8:21–23 (quoted at Pet. 63). According to Petitioner, this is accomplished using the “Last Bank” flip-flop 207, which Paluch discloses “identifies the bank of memory addressed by the most recent memory access performed by arbitrator 122.” Pet. 63;

Ex. 1006, 8:32–34. Dr. Tredennick testifies that “[t]he memory controller of Paluch then uses the information in flip flop 207 (which indicates the currently-active bank – i.e. current state) to compare to incoming requests to arbitrate requests between the two banks of the RAM.” Ex. 1002 ¶ 160. Based on this evidence, we find that Paluch’s disclosure of looking at the currently active memory bank of the requested memory teaches using a current state of the resource for a memory command.

As for the “requested state” of the resource, Petitioner cites the following disclosure of Paluch:

As part of generating addresses, address generator 124 determines whether an address delivered to DRAM interface 126 is the first address in a new page, thus will require that a new row address be delivered to memory 120. If a new row address must be delivered to memory 120, address generator 124 delivers a LOAD RAS signal on line 159. Once the row address has been delivered, subsequent accesses in the same memory page only require deliver of column addresses to the memory, and address generator 124 accordingly delivers a LOAD CAS signal on line 161 along with the address on line 158.

Ex. 1006, 5:5–14, *cited in* Pet. 64–65. We are persuaded, and we find, that Paluch’s disclosure of considering whether a new row needs to be addressed in memory to accommodate the memory access request (Ex. 1006, 5:5–14) teaches using a requested state of the resource, as recited in claim 1.

Petitioner also contends Hughes teaches using the current and requested states of the requested resource. Pet. 63–64 (citing Ex. 1008, 2:17–31; Ex. 1002 ¶ 161), 65–67 (citing Ex. 1008, 2:20–31, 5:66–6:5, 6:38–49, 7:1–15, Fig. 4; Ex. 1002 ¶¶ 165–168). For example, Hughes discloses that “[t]he memory controller includes logic responsive to the characteristics of the request or requests currently being executed in the shared memory

pipeline and to the characteristics of pending requests, to select a next request for access to improve pipeline fullness.” Ex. 1008, 2:25–30.

Referring to the process of Figure 4, Hughes discloses the following:

[T]he algorithm determines the length of the pending request at block 154; determines the beginning address of the pending request (block 155); and determines the request type of the pending request at block 156. *Using this information and the information about processes currently in the pipeline*, the algorithm selects first longer requests (block 157), then selects requests having consecutive addresses (block 150a), then selects requests of the same type (block 159), then selects the refresh process if it is near a refresh time out at block 160, and then selects the oldest remaining request among the set of requests after filtering by the previous parameters at block 161. The oldest remaining request is processed at block 153 and the process is completed.

Id. at 7:2–15 (emphasis added), *cited in* Pet. 66–67. We are persuaded, and we find, that Hughes’s disclosure of using information on current operations and pending requests so as to “select[] requests having consecutive addresses” (Ex. 1008, 7:1–15), for example, teaches using a current state, i.e., the state reflected by the current memory operation, and a requested state, i.e., the state reflected by the requested memory access. *See also* Ex. 1008, 5:66–6:5 (“Thus, the source of the data, the beginning address of the data, the size of the access and the direction of the access are all utilized in combination with information about current requests pending in the shared memory pipeline, in order to select a next optimum request according to the parameters of the particular implementation.”).

Petitioner further contends Paluch teaches “generat[ing] a corresponding sequenced command suitable for accessing the resource as required by the requesting system.” Pet. 68–69. According to Petitioner,

“[t]he memory controller of Paluch generates and sequences pre-charge, read, and write commands and determines the number of cycles between them.” Pet. 68. Referring to the operation of logic 480 in state machine 406 of DRAM interface 126, Paluch discloses that “state transition logic 480 has eight states through which it might pass. Each state of FIG. 4D is associated with the set of output values for the column address strobe, row address strobe and write enable signals on lines 164, 166 and 408.” Ex. 1006, 30:62–66. Based on this disclosure, we are persuaded, and we find, that Paluch teaches “generat[ing] a corresponding sequenced command suitable for accessing the resource as required by the requesting system.”

d. Rationale to Combine

Petitioner provides reasons that a person of ordinary skill in the art would have combined Paluch and Hughes in the manner asserted. Pet. 52–54, 67. For example, Petitioner contends that “Paluch teaches greater efficiency by switching accesses between various banks, and Hughes teaches greater efficiency by ordering requests in a queue” and that “the combination of these teachings results in a system with greater efficiency than either one alone.” *Id.* at 53–54 (citing Ex. 1002 ¶ 141). Dr. Tredennick testifies that “[i]t was known at the time that memory control was a bottleneck in the speed of computing, and thus skilled artisans would focus on improving the efficiency and speed of memory accesses to speed up the overall operation of computers.” Ex. 1002 ¶ 141. Dr. Tredennick also testifies that the teachings of Paluch and Hughes “are fully compatible and the combination of these teachings results in a system with greater efficiency than either one alone” and that “a skilled artisan would be motivated to combine the efficiencies taught by Hughes to the memory controller of Paluch to increase

computing speed and efficiency.” *Id.* We are persuaded by Petitioner’s reasoning, and we find a person of ordinary skill in the art would have had reason to combine the teachings of Paluch and Hughes to increase computing speed and efficiency, consistent with Dr. Tredennick’s testimony.

e. Conclusion as to claims 1 and 14

In summary, we find the combination of Paluch and Hughes teaches all of the limitations of claims 1 and 14. Patent Owner does not present any objective evidence of nonobviousness as to any of the challenged claims. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 1 and 14 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

4. Claims 2 and 15

Claims 2 and 15 depend, respectively, from claims 1 and 14 and recite that the “universal resource access controller” comprises various elements, which we discuss below.

a. Configurable system interface

Claims 2 and 15 recite “a configurable system interface coupled to the requesting system suitably arranged to both receive the resource access request and to generate a corresponding universal command.” Petitioner contends that Paluch’s memory controller teaches this limitation because arbitrator 122 receives access requests and the memory controller generates a corresponding command based on the request, as discussed for claim 1. Pet. 71–72 (citing Ex. 1006, 4:23–26, 4:30–5:15, Fig. 1; Ex. 1002 ¶¶ 71, 176–177). We are persuaded by Petitioner’s argument because Paluch discloses that “[a]rbitrator 122 stores each request” that is received from

functional units 102–112 and that “arbitrator 122 and address generator 124 interact to generate a sequence of multiple accesses of memory locations in response to a single request from a functional unit.” Ex. 1006, 4:23–26, 4:30.

Based on this evidence, therefore, we find Paluch teaches “a configurable system interface coupled to the requesting system suitably arranged to both receive the resource access request and to generate a corresponding universal command.”

b. Universal command sequencer

Claims 2 and 15 further recite “a universal command sequencer coupled to the configurable system interface.” Referring to its contentions for claim 1, Petitioner contends, and we are persuaded, that Paluch’s state machine 406 generates the sequenced command. Pet. 73–74 (citing Ex. 1002 ¶¶ 73, 179). In particular, Paluch discloses that state machine 406 in DRAM interface 126 “generates the column address strobe and row address strobe signals on lines 164 and 166 to read or write DRAM memory 120” and “produces a write enable signal on line 408 for directing DRAM memory 120 to read or alternatively to write at a specified memory address.” Ex. 1006, 23:40–45.

We find, therefore, that Paluch teaches “a universal command sequencer coupled to the configurable system interface.”

c. Resource tag buffer

Claims 2 and 15 further recite “a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current state of the requested resource and a resource tag arranged to identify the requested state of the requested resource.”

Petitioner contends Hughes discloses that pending requests are stored in blocks 104–107 of Figure 3. Pet. 74 (citing Ex. 1008, 5:40–42, Fig. 3). Petitioner reproduces the following excerpt of Hughes’s Figure 3:

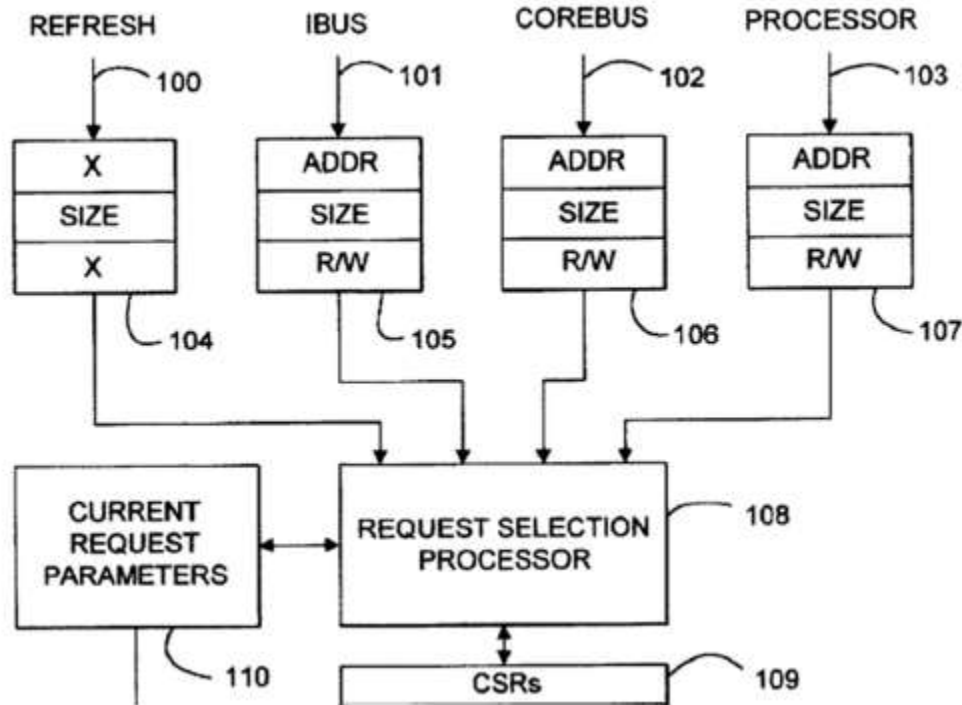


Figure 3 of Hughes is a diagram of shared memory arbiter/controller logic 72 showing four sources of access to the shared memory system—paths 100–103. Ex. 1008, 5:35–40. Petitioner cites Hughes’s disclosure that “requests which are currently in the shared memory pipeline are monitored in block 110.” Ex. 1008, 5:49–50, *cited in* Pet. 74. Petitioner further cites Hughes’s disclosure that the information in blocks 104–107 is used “in combination with information about current requests pending in the shared memory pipeline, in order to select a next optimum request.” Ex. 1008, 5:63–6:3, *cited in* Pet. 75.

Patent Owner argues that structures 104–107 and 110 in Hughes are different storage structures and, therefore, cannot be considered “a resource

tag buffer.” PO Resp. 40–46. Hughes describes Figure 3 as “a conceptual diagram of the shared memory arbiter/controller logic 72.” Ex. 1008, 5:35–36. Although Figure 3 of Hughes depicts the identified storage areas separately in arbiter/controller logic 72, we find that they comprise “a resource tag buffer” as claimed. As discussed above in the section addressing claim construction, the ’505 patent does not describe any particular structure for the resource tag buffer but, rather, describes the buffer in terms of what it stores, namely resource tags. See Section II.B.1 above. As discussed for claim 1, Hughes teaches using a current state and a requested state of a memory to fulfill memory requests. See Section II.F.3.c above. Hughes discloses that “[t]he memory controller includes logic responsive to the characteristics of the request or requests currently being executed in the shared memory pipeline and to the characteristics of pending requests, to select a next request for access to improve pipeline fullness.” Ex. 1008, 2:25–30. Hughes further explains how the current and requested states are used to process the next access request with respect to the flowchart of Figure 4, “which fundamentally illustrates the request selection process of the present invention.” Ex. 1008, 6:58–59. Thus, the disclosure of the request selection process in Figure 4 illustrates how request selection processor 108 in Figure 3 selects the next request. As discussed above for claim 1, Hughes discloses the following with reference to Figure 4:

[T]he algorithm determines the length of the pending request at block 154; determines the beginning address of the pending request (block 155); and determines the request type of the pending request at block 156. *Using this information and the information about processes currently in the pipeline*, the algorithm selects first longer requests (block 157), then selects requests having consecutive addresses (block 150a), then selects

requests of the same type (block 159), then selects the refresh process if it is near a refresh time out at block 160, and then selects the oldest remaining request among the set of requests after filtering by the previous parameters at block 161. The oldest remaining request is processed at block 153 and the process is completed.

Ex. 1008, 7:2–15 (emphasis added), *cited in* Pet. 66–67. Thus, Hughes describes using information on current operations and pending requests, and Hughes describes that “requests which are currently in the shared memory pipeline are monitored in block 110.” Ex. 1008, 5:49–50. Thus, given Hughes’s disclosure that “current request parameters” block 110 monitors the current memory requests, we find that block 110 stores a current state of the memory. We find that buffers 105, 106, and 107⁶ store requested states of memory because they store pending requests. Ex. 1008, 5:43–45 (“Each request is stored in a buffer represented by blocks 104 through 107 in the figure.”). We find that Hughes’s storage locations for the current and requested states of the requested resource teach “a resource tag buffer.”

Furthermore, Hughes teaches “a resource tag buffer” under Dr. Przybylski’s definition of “buffer.” Dr. Przybylski testifies, “To be viewed as single buffer by a person of ordinary skill, the unit must be logically or physically a single entity. Logical unity involves storing one or more values of the same type, typically with a common input and common output.” Ex. 2002 ¶ 38; *see also* Tr. 32:16–17 (Patent Owner’s counsel stating that a buffer “must, either logically or physically[,] be a unit”).

⁶ We focus on buffers 105, 106, and 107, rather than on buffer 104, because buffer 104 is in refresh path 100, and Hughes states that, “[i]n the refresh path 100, the address and read or write parameters are not utilized.” Ex. 1008, 5:42–43, Fig. 3.

Hughes describes such “logical unity.” As discussed above, Hughes teaches using a current state and a requested state of a memory to fulfill memory requests. Dr. Tredennick testifies that “the invention of Hughes involves comparing information stored about the current state and information stored about the requested state, meaning that these different storage locations have a ‘commonality of purpose’ and are ‘logically related.’” Ex. 1018 ¶ 11 (citing Ex. 1008, 3:18–20, 5:49–50, 5:66–6:3). We credit this testimony because, in its discussion of Figure 4, Hughes itself explains that certain address tables “include information which is used *when request candidates are compared to the current access.*” Ex. 1008, 7:65–66 (emphasis added). Thus, the state information for current access and requested access are compared, consistent with Dr. Tredennick’s testimony. Ex. 1018 ¶ 11. Therefore, the values of the requested state and the current state are of the same type so that Hughes’s request selection processor 108 can compare them. See Ex. 2002 ¶ 38 (“Logical unity involves storing one or more values of the same type . . .”). We, therefore, agree with Petitioner that “a comparison is being made using similar information about the current state and the requested state, meaning that structures for storing that information are ‘logically related’ according to Patent Owner’s proposed construction.” Pet. Reply 23.

Based on the foregoing, we find Hughes teaches “a resource tag buffer coupled to the command sequencer arranged to store a resource tag arranged to identify the current state of the requested resource and a resource tag arranged to identify the requested state of the requested resource.”

d. Characteristic operating parameter buffer

Claims 2 and 15 further recite “a characteristic operating parameter buffer coupled to the command sequencer arranged to store the characteristic operating parameter associated with the requested resource.” Petitioner refers to its contentions for the “characteristic operating parameter” limitation of claim 1. Pet. 75. There, Petitioner asserts Paluch’s DRAM timing registers 488 store timing specifications for the memory. Pet. 58–60. We are persuaded Paluch teaches this subject matter because it describes that “a collection of DRAM timing registers 488 is accessible to state transition logic 480. The DRAM timing registers identify the duration of time for which state transition logic 480 should remain in each of its output states as discussed below.” Ex. 1006, 30:21–24, *cited in* Pet. 59. We find Paluch’s timing registers teach “a characteristic operating parameter buffer coupled to the command sequencer arranged to store the characteristic operating parameter associated with the requested resource.”

e. Generating the sequenced universal command

Claim 2 further recites, “wherein the universal command sequencer uses the respective resource tags that identify the current state and the requested state of the requested resource and the characteristic operating parameter associated with the requested resource to generate the sequenced universal command.” Petitioner refers to its contentions for the limitation of claim 1 reciting “generat[ing] a corresponding sequenced command.” Pet. 75. For the reasons explained for claim 1, we find Paluch teaches this subject matter. *See* Section II.F.3.c above.

f. Conclusion as to claims 2 and 15

In summary, we find the combination of Paluch and Hughes teaches all of the limitations of claims 2 and 15. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 2 and 15 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

5. Claims 3 and 16

Claim 3 and 16, which depend from claims 2 and 14, respectively, recite, “wherein the requesting system is one of a plurality of processors included in a multiprocessor computing system, wherein the configurable system interface is configurable to be able to accept and process resource requests from any of the plurality of processors.” Petitioner contends Paluch’s functional units 102–110 teach a plurality of processors. Pet. 75 (citing Ex. 1006, Fig. 1; Ex. 1002 ¶ 185). We agree. Paluch discloses that “functional modules 102 through 112 . . . perform video decoding and audio decoding tasks for the purposes of digital video reception.” Ex. 1006, 3:27–29. In Figure 1, Paluch illustrates audio decoder 102, video decoder 104, output encoder 108, and CPU 110. Based on Paluch’s descriptions of these devices, we are persuaded they are processors. Ex. 1006, 3:50–4:2.

Petitioner contends “[b]oth Paluch and Hughes disclose receiving requests from multiple sources and processing those various requests.” Pet. 76–77 (citing Ex. 1008, Figs. 3, 4; Ex. 1006, 3:39–45; Ex. 1002 ¶ 186). We agree. Hughes describes that the “controller receives multiple requests” in step 150 of Figure 4. Paluch describes that its

memory controller includes an arbitrator 122 for determining which of the functional units 102 through 112 may access memory 120, an address generator 124 for generating addresses in response to requests for memory access from functional units 102 through 112, and a DRAM interface 126 for generating address and control signals for controlling DRAM 120.

Ex. 1006, 3:39–45. Thus, we find the combination of Paluch and Hughes teaches accepting and processing requests from a plurality of processors.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 3 and 16 would have been obvious under 35 U.S.C.

§ 103(a) based on the combined teachings of Paluch and Hughes.

6. *Claim 4*

Claim 4 recites,

A universal resource access controller as recited in claim 3, wherein the requested resource is one of a plurality of shared resources wherein each of the plurality of shared resources has associated with it a characteristic operating parameter which is stored in a corresponding operating characteristic parameter buffer and wherein each of the plurality of shared resources has associated with it a resource tag indicating a current state of each of the plurality of resources and a resource tag indicating a requested state of each of the plurality of resources.

Petitioner contends Hughes discloses controlling two memory devices as part of shared memory 25. Pet. 77 (citing Ex. 1008, 5:56–58; Ex. 1002 ¶ 187). We agree because Hughes discloses that “shared memory system 25 includes in this example a first tier 117-0 and a second tier 117-1.”

Ex. 1008, 5:56–58.

Petitioner refers to portions of its contentions for claims 1 and 2 for teachings of a “characteristic operating parameter buffer” and the recited resource tags. Pet. 77. For the reasons explained as to claims 1 and 2, we

are persuaded the combination of Paluch and Hughes teaches that each shared resource “has associated with it a characteristic operating parameter which is stored in a corresponding operating characteristic parameter buffer” and “a resource tag indicating a current state of each of the plurality of resources and a resource tag indicating a requested state of each of the plurality of resources.” See Sections II.F.3.c, II.F.4.c, and II.F.4.d. above.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 4 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

7. Claims 5–7

Claim 5 depends from claim 4 and recites, in part, “an address space controller coupled to the command sequencer arranged to store each of the operating characteristic parameter buffers associated with each of the plurality of shared resources.” Petitioner’s entire assertion as to this limitation is as follows: “The combination of Paluch and Hughes could and would be modified to allow Paluch’s DRAM timing registers to store information about each of a plurality of memory modules. See 4[b], *supra*; [Ex. 1002] ¶ 190.” Pet. 78. The cited testimony of Petitioner’s declarant is similarly conclusory. See Ex. 1002 ¶ 190. In our Decision on Institution, we stated that “Petitioner alleges that something could have and would have been done, but Petitioner neither provides a reason to do this nor directs us to evidence for why it would have been done.” Dec. on Inst. 29 (citing *In re Nuvasive*, 842 F.3d 1376, 1382 (Fed. Cir. 2016)). During the trial, Petitioner did not provide additional argument regarding claim 5 for this ground of unpatentability.

Having considered the full record developed during trial, we determine Petitioner has not shown, by a preponderance of the evidence, that claim 5 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes. Claims 6 and 7 depend from claim 5, and Petitioner's contentions for claims 6 and 7 do not remedy the deficiencies of Petitioner's contentions for claim 5. Therefore, Petitioner has not shown, by a preponderance of the evidence, that claims 6 and 7 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

8. Claim 11

Claim 11 is directed to “[a] multi-processor computer system” having “a system bus” and “a plurality of processors each capable of issuing a processor command and an associated data burst to the system bus.” Petitioner contends Paluch's data paths 140, 150, and 152 teach buses. Pet. 80–81 (citing Ex. 1006, 4:8–11, 4:12–19, Fig. 1; Ex. 1002 ¶ 196). Petitioner also contends Paluch teaches data bursts. Pet. 82 (citing Ex. 1006, 20:52–65; Ex. 1002 ¶ 198). We are persuaded by these contentions. As discussed for claim 3, we find Paluch teaches a system having a plurality of processors, namely functional modules 102, 104, 108, and 110. See Section II.F.5 above. Paluch explains that “each functional unit 102 through 112 delivers virtual address and additional memory access request information through a bus 150.” Ex. 1006, 4:12–14. Paluch further discloses that, “when memory 120 is formed of SDRAM memory chips (which are 16-bits in width), memory 120 is programmed to ‘burst’, i.e., read or write, two memory locations for each column address delivered thereto.” Ex. 1006,

19:56–60, *quoted in* Pet. 82.⁷ Thus, we find Paluch teaches “a plurality of processors each capable of issuing a processor command and an associated data burst to the system bus.”

Claim 11 further recites, “a memory controller as recited in claim 1 connected to the plurality of processors by way of the system bus, the memory controller being arranged to receive the processor commands and associated data bursts issued by the processors and to issue corresponding shared memory commands,” and “a shared memory device.” As discussed for claims 1 and 3, we find Paluch teaches a memory controller that is connected to a plurality of processors (Paluch’s “functional modules”) and receives processor commands and issues memory commands. As discussed for the previous limitations of claim 11, Paluch discloses write data bursts, which we find teach “data bursts issued by the processors.” Paluch also discloses a shared memory device, namely DRAM 120. Ex. 1006, Fig. 1.

Claim 11 further recites “a command bus connecting the shared memory device to the memory controller arranged to carry the issued shared memory commands according to a minimum issue time generated by the memory controller” and “a data bus connecting the shared memory to the memory controller arranged to carry read data from the shared memory to the memory controller based upon the data offset generated by the memory controller.” Petitioner contends that lines 162, 164, and 166 in Paluch’s Figure 1 teach the claimed “command bus” and that data bus 170 teaches the

⁷ Although Petitioner cites column 20, the material quoted by Petitioner appears in column 19.

claimed “data bus.” Pet. 83–84 (citing Ex. 1006, 5:45–49, 32:56–59,⁸ Fig. 1; Ex. 1002 ¶ 201), 86 (citing Ex. 1006, 5:49–52; Ex. 1002 ¶ 204). We agree. Paluch discloses that “DRAM interface 126 further generates column address strobe signals on line 164 and row address strobe signals on line 166 to cause DRAM 120 to access memory locations identified by address bits on bus 162.” Ex. 1006, 5:45–49. Paluch also discloses the need to wait to perform a memory access in certain circumstances. Ex. 1006, 32:56–59. Thus, we find Paluch teaches “a command bus connecting the shared memory device to the memory controller arranged to carry the issued shared memory commands according to a minimum issue time generated by the memory controller.” As for the “data bus” limitation, Paluch shows bidirectional bus 170 connecting DRAM 120 and DRAM interface 126, which is part of the memory controller, and Paluch discloses that data retrieved from DRAM 120 flow over bus 170. Ex. 1006, 5:51. Thus, we find Paluch teaches “a data bus connecting the shared memory to the memory controller arranged to carry read data from the shared memory to the memory controller based upon the data offset generated by the memory controller.”

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 11 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

⁸ Although Petitioner cites column 30, the material quoted by Petitioner appears in column 32. Pet. 84.

9. Claims 12 and 17

Claim 12 recites, “The multi-processor computer system as recited in claim 11, wherein the shared memory is an SDRAM.” Claim 17 recites, “A method as recited in claim 14, wherein the requested resource is a memory device.” As discussed for claim 11, Paluch discloses DRAM 120 as a shared memory device. Ex. 1006, 3:34–37 (“Each of the functional units 102 through 112 interacts in various ways with a large off-chip memory 120, which is a dynamic random access memory or DRAM.”), Fig. 1.

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claims 12 and 17 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

10. Claim 13

Claim 13 recites, “The multi-processor computer system as recited in claim 11, wherein the shared memory is a SLDRAM module having a plurality of SDRAMs.” Citing the testimony of Dr. Tredennick, Petitioner contends that “[i]t would have been obvious and easy to replace SDRAM with SLDRAM.” Pet. 87 (citing Ex. 1002 ¶¶ 116, 207). Dr. Tredennick testifies, “As discussed above in reference to Claim 13 in Ground 1, SLDRAM was designed to be a more-advanced and backwards compatible version of SDRAM (though it never made it to production.) Thus, it would have been obvious and easy to replace SDRAM with SLDRAM.” Ex. 1002 ¶ 207. With respect to claim 13 in the ground of unpatentability based on Chauvel, Dr. Tredennick testifies that “Synchronous Link Dynamic Access Memory was a proposed (but not ever implemented) enhanced version of

SDRAM that used a multiplexed bus to transfer data to and from the chips rather than fixed pin settings.” Ex. 1002 ¶ 114.

We find that SLDRAM’s improvements over SDRAM would have provided a reason for a person of ordinary skill in the art to use SLDRAM, and its backward compatibility would place such a modification well within the skill level of a person of ordinary skill in the art, as Dr. Tredennick testifies. Ex. 1002 ¶¶ 114–115. Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 13 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

11. Claim 18

Claim 18 recites, “A method as recited in claim 17, wherein the memory device is shared amongst a number of memory devices each of which are coupled to the universal resource access controller.” As discussed for claim 4, Hughes discloses controlling two memory devices as part of shared memory 25, namely a first tier device and a second tier device. Ex. 1008, 4:20–23, 5:56–58. Hughes discloses that “controller/arbitrator 20 is coupled to an SDRAM shared memory system 25 which includes a first tier SDRAM chip 26-1, and a second tier SDRAM chip 26-2.” Ex. 1008, 4:20–23. Hughes depicts DRAM 25 connected to DRAM controller 20 in Figures 1 and 2. Thus, we find the combination of Paluch and Hughes teaches that “the memory device is shared amongst a number of memory devices each of which are coupled to the universal resource access controller.”

Having considered the full record developed during trial, we conclude Petitioner has shown, by a preponderance of the evidence, that the subject matter of claim 18 would have been obvious under 35 U.S.C. § 103(a) based on the combined teachings of Paluch and Hughes.

*G. Obviousness over Paluch and Chauvel
(Claims 1–7 and 11–18)*

Petitioner also contends claims 1–7 and 11–18 of the ’505 patent are unpatentable under 35 U.S.C. § 103(a) as having been obvious over the combined teachings of Paluch and Chauvel. Pet. 1–2, 51–87. Petitioner presents this ground in the same part of its Petition as the ground based on Paluch and Hughes and introduces these grounds by stating that “the specifics of each combination—Paluch with Hughes and Paluch with Chauvel—are discussed” below. Pet. 54–55. As noted on page 30 of our Decision on Institution, however, Petitioner does not rely on Chauvel at all in its discussion of claims 2 and 15 in these grounds. See Pet. 71–75. Indeed, in this ground, Petitioner relies expressly on Chauvel only once, referring to Chauvel’s teachings for the use of a requested state of the requested resource in claim 1. Pet. 67–68. Thus, with only a single mention of Chauvel in these grounds, Petitioner’s contentions for these grounds are based primarily on Paluch and Hughes. We have already determined that claims 1–18 would have been obvious over Chauvel, and we have also determined that claims 1–4 and 11–18 would have been obvious based on the combined teachings of Paluch and Chauvel. Thus, this Decision addresses all claims challenged and several claims under multiple grounds. See 35 U.S.C. § 318(a) (“If an inter partes review is instituted and not dismissed under this chapter, the Patent Trial and Appeal Board shall issue a final written decision with respect to the patentability of any patent claim

challenged by the petitioner and any new claim added under section 316(d).”). Furthermore, this Decision resolves the disputes between the parties, which focus entirely on “resource tag buffer” in claims 2 and 15. Given that Petitioner does not rely on Chauvel for any teachings for claims 2 and 15 in this ground (based on Paluch and Chauvel) and, thus, there is no additional dispute to resolve between the parties, we decline to assess separately this asserted ground.

III. MOTION TO EXCLUDE

Patent Owner filed a Motion to Exclude Exhibit 1012, which is an excerpt from Patent Owner’s Preliminary Infringement Contentions in co-pending district court litigation. Paper 25. We do not rely on Exhibit 1012, and therefore, we dismiss Patent Owner’s Motion to Exclude as moot.

IV. CONCLUSION

For the reasons discussed above, we determine Petitioner has proven, by a preponderance of the evidence, that (1) claims 1–18 are unpatentable under 35 U.S.C. § 103(a) over the teachings of Chauvel; (2) claims 3–13, 16, and 18 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Chauvel and Ramanujan; and (3) claims 1–4 and 11–18 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Hughes. Petitioner, however, has not proven, by a preponderance of the evidence, that claims 5–7 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Paluch and Hughes.

V. ORDER

Accordingly, it is:

ORDERED that claims 1–18 of the '505 patent have been shown to be unpatentable;

FURTHER ORDERED that Patent Owner's Motion to Exclude Exhibit 1012 is dismissed as moot; and

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

IPR2017-01500
Patent 6,532,505 B1

PETITIONER:

David Hoffman
hoffman@fr.com

Jeremy Monaldo
jjm@fr.com

W. Karl Renner
axf-ptab@fr.com

PATENT OWNER:

Matthew Phillips
mphillips@lpiplaw.com

Derek Meeker
derek.meeker@meekerip.com

Kevin Laurence
klaurence@lpiplaw.com

Bryan Richardson
brichardson@wilan.com