

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ARM LTD¹,
Petitioners,

v.

DAEDALUS PRIME LLC,
Patent Owner.

IPR2025-00207
Patent 8,984,228 B2

Before, GREGG I. ANDERSON, ARTHUR M. PESLAK and
MARY C. HOFFMAN, *Administrative Patent Judges*.

ANDERSON, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

¹ MediaTek Inc. was originally a petitioner but as a result of settlement was terminated from this proceeding. *See* Paper 29.

I. INTRODUCTION

A. Background

Arm Ltd. (“Petitioner”) filed a Petition requesting *inter partes* review of claims 1–7, 11–14, and 16–19 (the “challenged claims”) of U.S. Patent No. 8,984,228 B2 (Ex. 1001, the “’228 patent”). Paper 3 (“Pet.”). Petitioner filed a request for Director Review of the Director’s Decision denying institution (“Decision,” Paper 10) and Patent Owner filed an authorized response. *See* Paper 12 (“DR Request”); Paper 13 (“Patent Owner Response”). The Director granted the DR Request and referred the Petition to the Board “to handle the case in the normal course, including by issuing a decision on institution addressing the merits and other non-discretionary considerations, as appropriate.” Paper 14 (“Order Granting Director Review”).

Daedalus Prime LLC (“Patent Owner”) waived “filing of a preliminary response on the merits.” Paper 8. We instituted *inter partes* review. Paper 15 (“Inst. Dec.”). Patent Owner filed a Response (Paper 23, “PO Resp.”), Petitioner filed a Reply (Paper 25, “Pet. Reply”), and Patent Owner filed a Sur-Reply (Paper 26, “PO Sur-Reply”). A hearing was held June 25, 2026, and a transcript has been made of record. Paper 38 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6. This decision is a Final Written Decision issued pursuant to 35 U.S.C. § 318(a). For the reasons we discuss below, we determine that Petitioner has proven by a preponderance of the evidence that all of the challenged claims of the ’228 patent are unpatentable.

B. Related Proceedings

The parties identify the following district court proceeding involving the '228 patent: *Daedalus Prime LLC v. MediaTek Inc.*, No. 2:24-cv-00235-JRG (E.D. Tex.) (Ex. 1019). Pet. 81; Paper 4 (“Patent Owner’s Mandatory Notices”), 2.

In Petitioner’s request for Director Review, Petitioner identified *Mercedes-Benz USA, LLC v. Daedalus Prime LLC*, IPR2023-01344 (“’1344 IPR”), where in a final decision the Board cancelled “substantially identical claims in the child patent^[2] to the challenged ’228” patent.” DR Request 2; *see* ’1344 IPR, Paper 31 (“’1344 IPR Final Decision”). Per our order at the hearing, both parties filed briefs regarding the effect of the ’1344 IPR Final Decision on this proceeding. *See* Tr. 38:11–23; Petitioner Arm Ltd.’s Issue Preclusion Brief (Paper 34, “Pet. Br.”); Patent Owner’s Reply Brief Regarding the Effect of the FWD of IPR 2023-01344 (Paper 37, “PO Reply Br.”); Patent Owner Supplemental Brief Regarding the Effect of the FWD of IPR2023-01344 (Paper 35, “PO Br.”); Petitioner Arm Ltd.’s Issue Preclusion Reply Brief (Paper 36, “Pet. Reply Br. ’1344”).

C. Real Parties-in-Interest

Petitioner identifies itself as the real party-in-interest. Pet. 81. Patent Owner identifies itself as the real party-in-interest. Paper 4, 2.

² US 9,575,895 B2 (“’895 patent”) in the ’1344 IPR is a continuation of the application for the ’228 patent. ’1344 IPR, Ex. 1001 at code [63].

D. The '228 Patent

The '228 patent was filed December 13, 2011. Ex. 1001, code (22). The application for the '228 patent was published June 13, 2013, as US 2013/0151782 A1. *Id.* at code (65).

1. Technical Background

The '228 patent relates to multiprocessor (MP)-socket computer systems, which may include different “semiconductor components realized as integrated circuits (ICs).” Ex. 1001, 1:6–7. “The ICs include processors, memories, chipsets, input/output hubs (IOHs).” *Id.* at 1:10–11.

Problems arise once an input/output (IO) component is integrated on the same chip with a multiprocessor because the IO has “a separate caching agent.” Ex. 1001, 1:19–21. This means a “dedicated logic is associated with the IO component to handle cache coherency operations.” *Id.* at 1:21–23. “When an IO agent is performing read/write operations to main memory, it has to snoop the CPU side cache to maintain cache coherency. In MP systems, this becomes a major scaling problem.” *Id.* at 1:23–26. “For example, in an 8 socket system, there are effectively 16 caching agents in the system, which can degrade performance.” *Id.* at 1:26–28.

2. The Common Caching Agent of the '228 Patent

The '228 patent describes how an “IO agent can be integrated into a multiprocessor socket such as a multicore processor.” Ex. 1001, 1:53–54. The scaling problem is addressed by configuring a CPU caching agent “to support both CPU traffic and IO traffic as well.” *Id.* at 1:55–58. “Performance can be enhanced by bringing the IO agent and CPU core(s) closer.” *Id.* at 1:58–59.

Figure 1 is reproduced below.

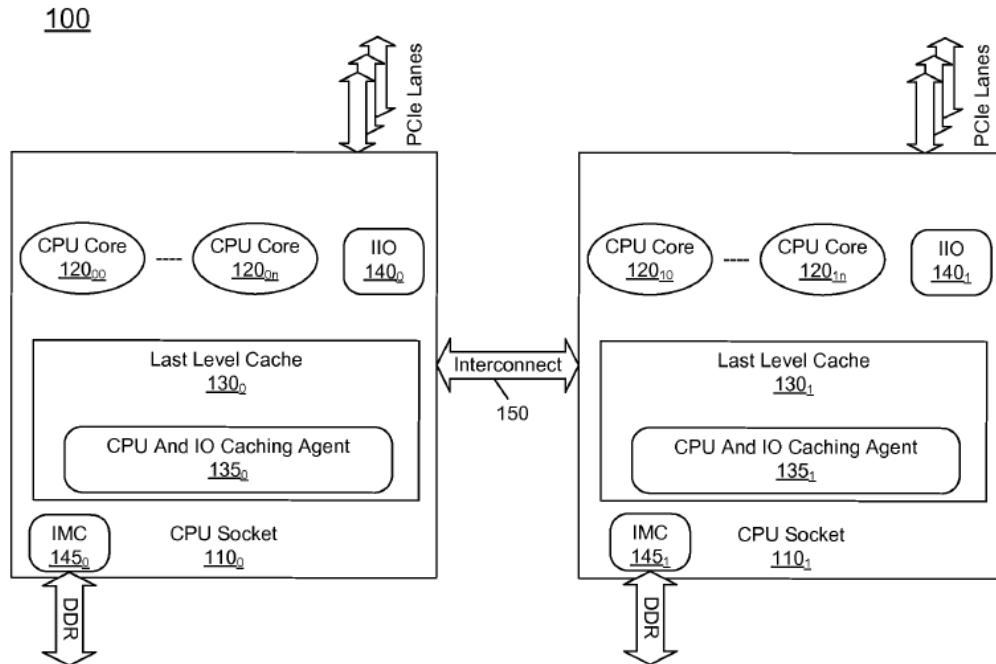


FIG. 1

Figure 1 is a block diagram of a portion of a multiprocessor system.

Ex. 1001, 1:34–35. As shown in Figure 1, system 100 includes a pair of processor sockets 110₀ and 110₁. *Id.* at 1:66–2:6.

[S]ocket 110₀ may include an IO module (hereafter an integrated IO module or IIO module). This IIO module may act as an interface to one or more off-chip peripheral or IO devices (not shown in FIG. 1) coupled to the processor, e.g., via PCIe™ links. Examples of such IO devices include storage devices, network interfaces, graphics cards, and so forth.

As further seen, socket 110₀ may include an integrated memory controller (IMC) 145₀ to provide an interface to a system memory (not shown for ease of illustration) via a memory interconnect. Socket 110₁ includes similar components and can

be coupled to socket 110₀ via an interconnect 150, which in one embodiment can be a point-to-point (PtP) link in accordance with a Intel® Quick Path Interconnect (QPI) protocol.

Id. at 2:21–35.

Figure 6 is reproduced below.

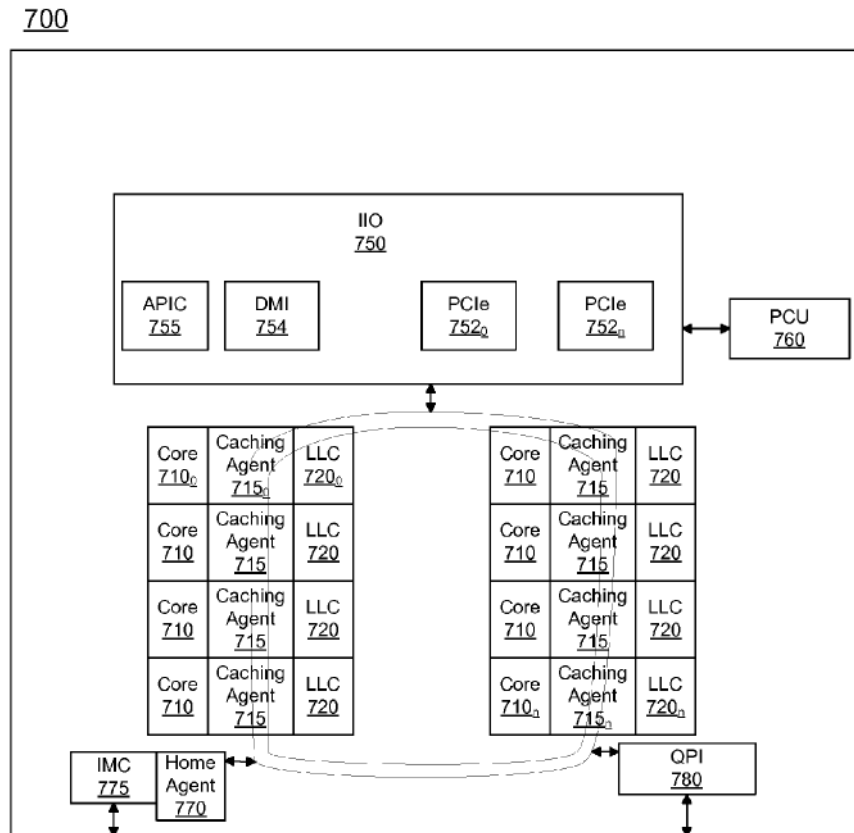


FIG. 6

Ex. 1001, 6:20–22. Figure 6 is a block diagram showing a “processor 700 includes a distributed configuration having partitions or slices each including a core 710 and a partition of a caching agent 715 and a LLC 720.” *Id.* at 6:23–26. The “distributed portions form a single caching agent, and which is configured to handle cache coherency operations both for the cores as well as an I/O module 750. *Id.* at 6:26–29.

E. Illustrative Claim

The Petition challenges claims 1–7, 11–14, and 16–19, of which claims 1, 11, and 16 are independent and directed to an apparatus, a method, and a system, respectively. Claim 1 is illustrative of the claimed subject matter and reproduced below:³

1. [1pre] An apparatus comprising:

[1a] a multicore processor including a plurality of cores,

[1b] a shared cache memory,

[1c] an integrated input/output (IIO) module to interface between the multicore processor and at least one IO device coupled to the multicore processor and

[1d] a caching agent to perform cache coherency operations for the plurality of cores and the IIO module

[1e] the caching agent a single caching agent for the multicore processor and including a plurality of distributed portions each associated with a corresponding one of the plurality of cores.

Ex. 1001, 7:36–46.

F. References and Other Evidence

The Petition relies on the following references:

Name⁴	Reference	Issue or Publication Date	Exhibit No.
Agarwal	US 7,805,575 B1	Sept. 28, 2010	Ex. 1005

³ Paragraph labeling in brackets is based on the labeling provided by Petitioner. *See, e.g.*, Pet. 10 (1[pre]).

⁴ First named author or inventor only.

Name ⁴	Reference	Issue or Publication Date	Exhibit No.
Sinharoy	<i>IBM POWER7 multicore server processor</i> , IBM Journal of Research and Development, Vol. 55, Issue 3, Pages 1:1–29	May/June 2011	Ex. 1006
Blumrich	US 2011/0219188 A1	Sept. 8, 2011	Ex. 1007
Sistla	US 2008/0126750 A1	May 29, 2008	Ex. 1008

In addition, Petitioner submits the Declaration of Marwan Hassoun, Ph.D., in support of the Petition (Ex. 1003, “Hassoun Declaration”). Patent Owner submits the Declaration of Nader Bagherzadeh, Ph.D., in support of Patent Owner’s Response (Ex. 2006, “Bagherzadeh Declaration”).

G. Asserted Grounds of Unpatentability

Petitioner asserts the challenged claims are unpatentable on the following grounds:

Ground	Claims Challenged	35 U.S.C. § ⁵	References
1	1, 2, 4–7, 11–14	103(a)	Agarwal
2A	1, 4, 6, 7, 11–14	103(a)	Sinharoy
2B	2, 3, 16–18	103(a)	Sinharoy, Blumrich

⁵ Because the earliest application from which the ’228 patent claims priority was filed before March 16, 2013, the pre-AIA (“America Invents Act”) version of § 103 applies. Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112-29, 125 Stat. 284, 285–88 (2011).

Ground	Claims Challenged	35 U.S.C. § ⁵	References
2C	5	103(a)	Sinharoy, Sistla
2D	19	103(a)	Sinharoy, Sistla, Blumrich

See Pet. 5 (listing Grounds).

H. Overview of the Asserted Prior Art

1. Agarwal (Ex. 1005)

Agarwal discloses a plurality of multicore processor and memory interfaces with access paths from cache memory to a main memory. Ex. 1005, code [57]. Agarwal teaches that “[c]ache coherence is a useful mechanism in multiple processor systems” because “[w]hen multiple processors cache a given shared memory location, a cache coherence problem may arise because a copy of the same memory location exists in multiple caches.” *Id.* at 1:34–39. “Directory based protocols are one way of maintaining cache coherence” using a “directory state and directory controllers.” *Id.* at 1:46–53.

“The directory can be located at a variety of places.” Ex. 1005, 11:12. For example, the directory controller and directory storage can be located on a node, e.g., a core or tile. *Id.* at 11:22–26. “The directory can be associated with that node’s higher level cache or the directory can be standalone by itself. This second scheme is called proxy cache coherence or PCC.” *Id.* at 11:23–26.

Figure 2 of Agarwal is reproduced below.

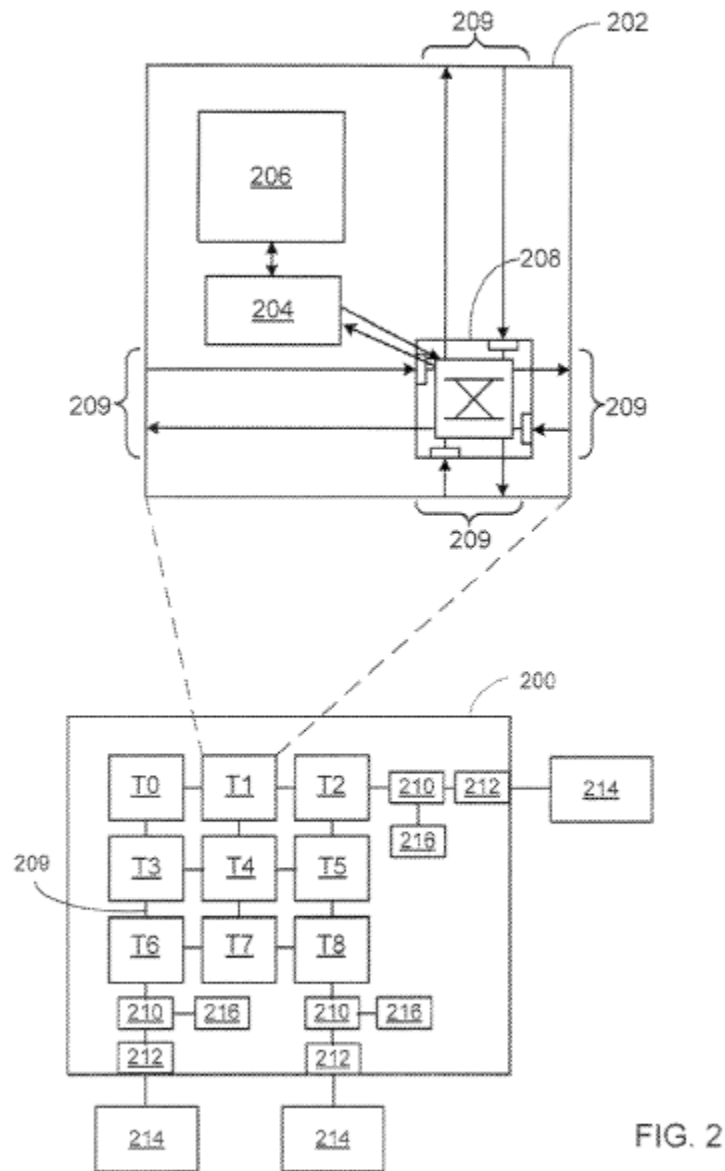


Figure 2 illustrates a Shim Cache Coherence (SCC) of an exemplary multicore processor 200 having a tiled processor architecture.

Ex. 1005, 11:50–53. “The SCC scheme applies to a wide range of multicore processor architectures as well as to multiprocessors in general.” *Id.* at 11:51–53. As shown in Figure 2, nine (9) “tiles (corresponding to the ‘processor cores’ or simply ‘cores’) numbered T0–T8, each tile 202

containing a processor 204 (or ‘processing engine’), a cache 206 and a switch 208.” *Id.* at 11:54–57.

Figure 5 of Agarwal is reproduced below.

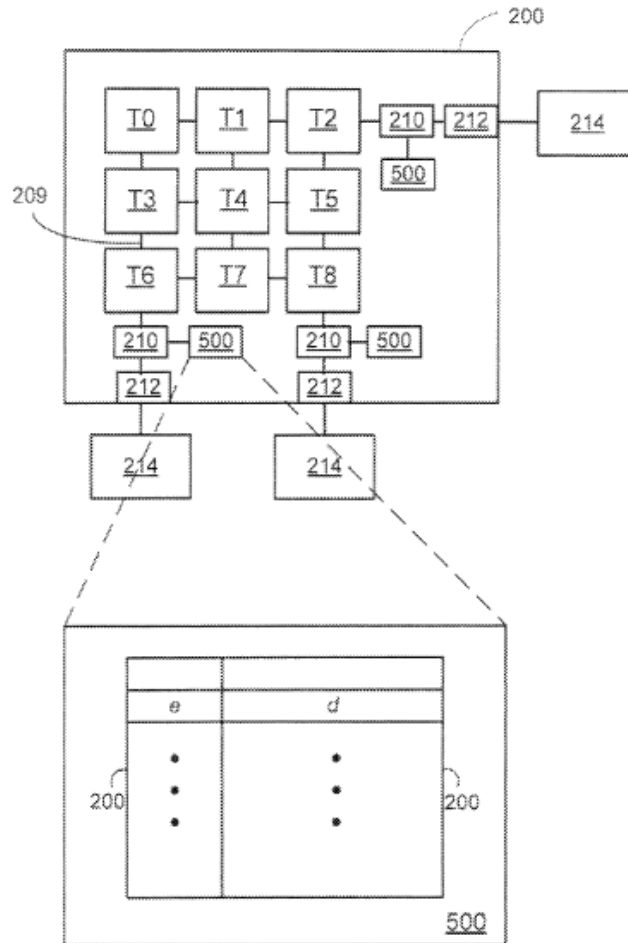


FIG. 5

Figure 5 shows a multicore processor 200 which includes a cache 500 associated with the shim 210 that stores data and directory entries.

Ex. 1005, 13:56–58. Figure 5 is an example where “the cache 500 can serve as an L3 data cache, while each tile includes an L1 and L2 cache.” *Id.* at 13:58–60.

Figure 6A of Agarwal is reproduced below.

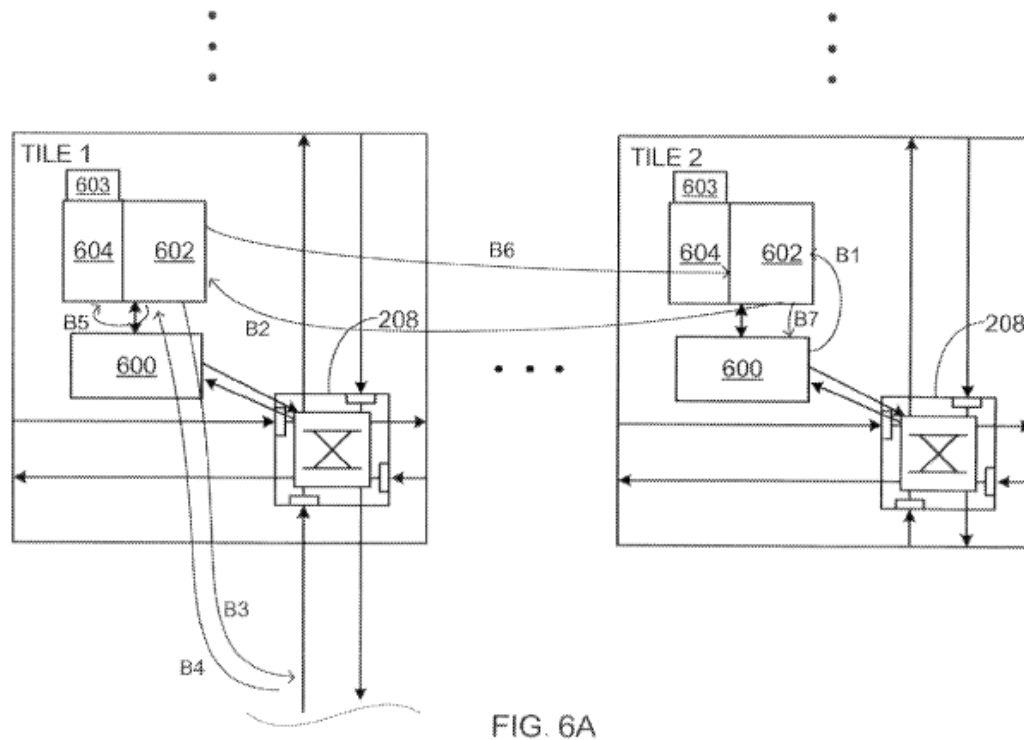


Figure 6A illustrates a Proxy Cache Coherence (PCC) of a tiled processor architecture.

Ex. 1005, 16:27–29.

Referring to FIG. 6A^[6], each of the tiles (e.g., tile 1 and tile 2) that are used as home tiles for a given portion of main memory includes a processor 600 coupled to a cache 602 (e.g., providing an L2 cache to back an L1 cache that is integrated into the processor 600). A directory storage 604 for directory state can be included within the same memory that provides the cache 602, or can be included in a separate memory module on the tile. In some implementations, directory entries can be stored alongside

⁶ Petitioner reproduces and describes Figure 6B, which illustrates a virtually identical architecture to Figure 6A. Pet. 8–9. *Compare* Ex. 1005, Fig. 6A, with Ex. 1005, Fig. 6B. Because Figure 6B is not specifically described in Agarwal, we reference Figure 6A.

cache tags for a given memory line. A directory controller 603 is coupled to the directory storage 604.

Id. at 16:35–45.

Figure 7C of Agarwal is reproduced below.

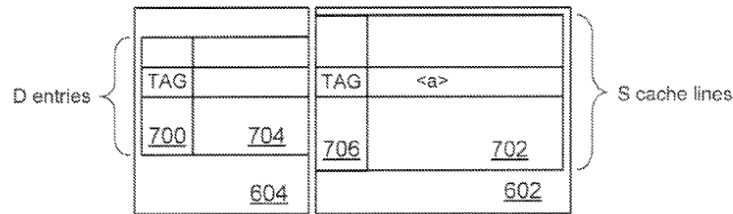


FIG. 7C

Figure 7C shows an example of an arrangement of a directory storage 604 in a memory providing a cache 602 of a tile.

Ex. 1005, 17:9–11. “Various techniques can be used to assign a given tile as the home location (e.g., home tile) for a given memory line (e.g., having address a),” as shown in Figure 7C. *Id.* at 17:42–44. “The home tile assigned to a given address a [see TAG in Fig. 7C] can also be based on a function of the address such as a hash function.” *Id.* at 17:50–51. For a shared distributed cache, “each memory line has an assigned home location in the cache 206” of one of the tiles. *Id.* at 23:51–53.

2. Sinharoy (Ex. 1006)

Sinharoy discloses a POWER7 server processor, which has eight cores, two memory controllers, an L2 cache, and an L3 cache. Ex. 1006, 2, 18–19. Sinharoy discloses that the POWER7 processor has a “256-KB POWER7 L2 cache” and a “shared 32-MB POWER L3 cache,” “composed of eight 4-MB L3 regions.” *Id.* at 18–19, Fig. 1. The L2 cache and L3 cache each has a single centralized controller to manage the read/write

request operations. *Id.* at 18–19. The POWER7 processor supports “two integrated I/O controllers.” *Id.* at 22. Each of the I/O controllers supports a “4-byte off-chip read interface and a 4-byte off-chip write interface, thereby connecting the POWER7 chip to up to two I/O hub chips.” *Id.* The I/O hub chips, PCI bridge, and PCI adapter are examples of input/output (I/O) devices that can be connected to the POWER7 processor via the integrated I/O controllers. *Id.* at 26, Fig. 14.

3. *Blumrich (Ex. 1007)*

Blumrich discloses a “multiprocessor system” methods to “maintain cache coherence.” Ex. 1007 at code [57], ¶ 124. Blumrich’s “L2 caches serve as the point of coherence for all processors.” *Id.* ¶ 119. As part of “[c]oherence tracking,” Blumrich uses a “request queue 302,” which “buffers incoming read and write requests.” *Id.* ¶¶ 127–128. In one embodiment, this queue “is 16 entries deep,” although “other request buffers might have more or less entries.” *Id.* Using Blumrich’s queue, “[t]he addresses of incoming requests are matched against all pending requests to determine ordering restrictions.” *Id.*

4. *Sistla (Ex. 1008)*

Sistla discloses a “processor” comprising “a memory,” “a plurality of processor cores in communication with the cache memory,” and “a scalability agent unit that operates as an interface between an on-die interconnect and both multiple processor cores and memory.” Ex. 1007, code (57).

II. ANALYSIS OF THE CHALLENGED CLAIMS

A. Level of Ordinary Skill in the Art

Petitioner contends to be a person of ordinary skill in the art of the '228 patent

would have (1) at least a bachelor's degree in electrical engineering, computer engineering, physics, or a similar field, and (2) approximately two years of industry or academic experience in a field related to microelectronic architecture design, including experience with cache management in multicore systems-on-a-chip (SoCs). (Ex-1003 ¶¶50-51.) More related education would compensate for fewer years of work experience (and vice versa). (Ex-1003 ¶¶50-51.)

Pet. 4–5.

This definition was adopted in the Institution Decision. Inst. Dec. 13. Patent Owner does not dispute the finding. PO Resp. 5–6. For the purpose of our Decision, we maintain the finding made in the Institution Decision.

B. Claim Construction

We construe claim terms only as relevant to the parties' contentions and only to the extent necessary to resolve the issues in dispute. *See Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999); *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

Petitioner contends that:

Given the similarities between the prior art and the '228 patent, no terms require construction to resolve this Petition. Related litigations may involve controversies not presented here that require construction.

Pet. 5.

1. “a single caching agent” (claim 1)

We note a construction issue from the ’1344 IPR where we construed a “single caching agent,” which is recited in claim 1 here. ’1344 IPR Final Decision, 16–18. We concluded that a “‘single caching agent’ means ‘one caching agent for the processor, which may be formed from distributed portions.’” *Id.* at 18. We adopted that construction in our Institution Decision in this case. Inst. Dec. 15.

The application for the ’985 patent in the ’1344 IPR is a continuation of the application for the ’288 patent. ’1344 IPR, Ex. 1001, code (63). “The prosecution history of a related patent can be relevant if, for example, it addresses a limitation in common with the patent in suit.” *See Advanced Cardiovascular Sys., Inc. v. Medtronic, Inc.*, 265 F.3d 1294, 1305 (Fed. Cir. 2001). As it relates to “a single caching agent,” the claim language between the two patents is substantially the same.⁷

Patent Owner argues that while

a “single caching agent” means “one caching agent for the processor,” that is not the whole story. The single caching agent should be understood as a logically unified entity, that is, a cohesive caching agent where all distributed portions are interconnected, consistent, and function together in a rational or

⁷ The claim in the ’985 patent recites, in relevant part, “wherein the caching agent is *a single caching agent for the processor* and includes a plurality of distributed portions each associated with a corresponding one of the plurality of cores.” ’1344 IPR, Ex. 1001, 7:48–61 (emphasis added). Claim 1 at issue here recites, again in relevant part, “the caching agent [is] *a single caching agent for the multicore processor* and including a plurality of distributed portions each associated with a corresponding one of the plurality of cores.” Ex. 1001, 7:43–46 (emphasis added).

systematic manner, not a collection of independent caching agents that merely communicate with one another.

PO Resp. 6 (citing Bagherzadeh Dec. ¶ 33).

Petitioner asserts Patent Owner proposes an “exclusionary meaning of distributed portions of the caching agent.” Pet. Reply 2. According to Petitioner, “the claims manage the overall shared cache, not individual pieces of that cache.” *Id.* Petitioner notes that “[c]laim 4 requires that the managed shared cache portions be associated with each core.” *Id.* at 2–3. Petitioner argues Agarwal’s “distributed cache coherency components manage in a cooperative fashion to achieve the coherency of the overall shared cache.” *Id.* Petitioner also cites Figure 6 of the ’228 patent as illustrating “the claimed ‘single caching agent’ with distributed portions, and cores and distributed slices / partitions of the shared cache / LLCs (memory regions) all associated with each other . . .”. *Id.* at 3.

In Patent Owner’s Sur-Reply, it is argued

The claims are not satisfied by a collection of independent controllers that exchange messages under a distributed protocol. They require a single, logically unified caching agent that performs coherency operations for all cores (and the IIO module), even if that agent is physically partitioned into distributed portions.

PO Sur-Reply 5 (citing Bagherzadeh Dec. ¶¶ 33, 39–44).

Petitioner argues that the distributed portions are managing the multicore processor together, “each associated with a corresponding one of the plurality of cores.” Tr. 4:15–22. Further, the cache agent is a logical construct. *Id.* Patent Owner agrees that the “single caching agent” is a “logically unified entity.” *Id.* at 19:7–8. The parties “disagree about what it

means for the distributed portions and the associated with language.” *Id.* at 19:8–9. This disagreement is addressed in Section II.E.1.f below.

During prosecution of the application for the ’228 patent, the applicant argued that the prior art did not disclose or suggest “a single caching agent.” Ex. 1002 (file history of ’228 patent), 42.⁸ The description of Figure 6 in the ’228 patent explains that a “single caching agent” may be formed from distributed portions. Ex. 1001, 6:25–29.

In construing the term, we have considered the arguments made relating to limitation 1e. *See* Section II.E.1.f below. We have also analyzed the effect of the ’1344 Final Decision, which construed “single caching agent,” as well as claim 1’s recitation of “distributed portions” where each is “associated with a corresponding one of the plurality of cores [“associated” limitation].” *See* Section II.C below. We find that our preliminary construction of “single caching agent” is supported on the final record and address the “associated” limitation in Sections II.C and II.E.1 below.

2. Remaining claim terms

Independent claims 11 and 16 include different language as to “caching agent.” Claim 11 recites “a caching agent.” Ex. 1001, 8:18. Claim 16 similarly recites “a first caching agent.” *Id.* at 8:47–48. Neither claims 11 or 16 includes claim 1’s recitation of “distributed portions.

The “caching agent” limitations of claims 11 and 16, are not at issue. Accordingly, no construction is necessary.

⁸ We use Petitioner’s page citations for Exhibit 1002.

C. Issues Relating to the '1344 IPR

The '228 patent is the parent of the '985 patent, which was the challenged patent in the '1344 IPR. *See* n.2 above. The grounds relating to Agarwal here were not presented in the '1344 IPR. Thus, we are only considering the Sinharoy grounds for purposes of whether issue preclusion applies.

As noted in Section I.B above, the parties have briefed the effect of the '1344 IPR Final Decision on this proceeding. As discussed below, the question of how the '1344 Final Decision applies here revolves around the “associated” limitation. Simply stated, Patent Owner argues the recited “distributed portions” are not the same as the recited “single caching agent.” *See* PO Resp. 10–12.

1. Petitioner's Argument

Petitioner argues “patent claims need not be identical for issue preclusion to apply Instead, the question is whether the differences between the claims ‘materially alter the question of invalidity.’ . . . these patents use slightly different language to describe substantially the same invention.” Pet. Br. 1–2 (citing *Apple Inc. v. Smart Mobile Techs. LLC*, No. 2024–1352, 2026 WL 160563, at *3 (Fed. Cir. Jan. 21, 2026) (quoting *Ohio Willow Wood Co. v. Alps S., LLC*, 735 F.3d 1333, 1342–43 (Fed. Cir. 2013))).

Petitioner quotes *Google LLC v. Hammond Dev. Int’l, Inc.*, 54 F.4th 1377, 1381 (Fed. Cir. 2022) (citations omitted), holding that issue preclusion applies when

- (1) the issue is identical to one decided in the first action; (2) the issue was actually litigated in the first action; (3) resolution of the issue was essential to a final judgment in the first action; and

(4) plaintiff had a full and fair opportunity to litigate the issue in the first action.

Pet. Br. 2. Further, Petitioner alleges the '1344 Final Decision was not appealed by Patent Owner and is therefore a basis for issue preclusion. *Id.* (citing *Power Integrations, Inc. v. Semiconductor Components Indus., LLC*, 926 F.3d 1306, 1311–12 (Fed. Cir. 2019)).

Petitioner argues the claim construction issue regarding “single caching agent” was decided in the '1344 Final Decision and Patent Owner is estopped from challenging it. Pet. Br. 3–4. We have independently considered the construction of that term and find that the construction from the '1344 IPR Final Decision applies here. *See* Section II.B.1 above; Sections II.C.2 and 3 below. Petitioner argues the claim construction is dispositive of the ground based on Sinharoy because it is Patent Owner's only argument. *Id.* at 6.

Petitioner produced a table showing the claims at issue here are “substantially the same” as corresponding claims of the '985 patent. Pet. Br. 6–7. Petitioner's table lists the claims in the '985 patent determined to be either anticipated or obvious based on the Sinharoy reference. *Id.* (citing '1344 Final Decision 19–31).

2. Patent Owner's Argument

Patent Owner does not dispute the applicable law as set forth by Petitioner. *See*, generally PO Br. As discussed below, Patent Owner also does not dispute our now final construction of “single caching agent.” *Id.* at 2. What Patent Owner does dispute is the “associated” limitation by arguing that the issue here is different, i.e., not “identical” for purposes of

issue preclusion, from what was decided in the '1344 Final Decision. PO Br. 2–3; *see also* PO Reply Br. 3; '1344 Final Decision 17 (“associated” limitation is a “separate limitation”).

Patent Owner also argues that Petitioner is broadening issue preclusion to cover claims that are different from claim 1. PO Reply Br. 4–5. For example, claims 11 and 16 here are not “identical” to claim 1. *Id.*

3. Discussion and Conclusion

Patent Owner does not ask us to revisit our construction of “single caching agent.” PO Reply Br. 2. For reasons set out previously, we maintain our construction of “single caching agent” from the Institution Decision. *See* Section II.B.1 above.

Concerning the Sinharoy ground, analyzed in Section II.F below, Patent Owner alleges the '1344 Final Decision did not reach the “associated” limitation of claim 1. PO Reply Br. 3–4 (citing '1344 Final Decision 17). Petitioner argues we did reach the “associated” limitation. Pet. Reply Br. 4–7. Petitioner also argues the “associated” limitation argument is a new argument that was waived. *Id.* at 7.

Patent Owner argues that the issues surrounding the “associated” limitation were not analyzed in the '1344 Final Decision. PO Reply Br. 3. Although in the '1344 Final Decision we specifically found the “associated” limitation was a “separate limitation,” that is not dispositive of the inquiry. In the '1344 Final Decision, we found Sinharoy anticipated or rendered obvious claims 1, 3–7, 11–13, and 15–17 of the '895 patent. '1344 Final Decision 42. As previously noted, claim 1 of the '228 patent and claim 1 of the '895 patent are substantially identical, including the “associated”

limitation. *See* n.7 above; *see also* Pet. Br. 1 (comparing claim 1 of the '895 patent to claim 1 of the '228 patent). Patent Owner seemingly accepts that claim 1 of the '228 patent is subject to issue preclusion and would have been obvious over Sinharoy for the reasons stated in the '1344 Final Decision. *See* PO Reply Br. 4–5 (arguing against extending issue preclusion to claims 11 and 16).

Here, we independently determine that Sinharoy renders obvious claims 1, 4, 6, 7, and 11–14. *See* Section II.F below. Accordingly, we need not rely on issue preclusion.

Notwithstanding the preceding, the necessary elements to find issue preclusion on the Sinharoy grounds are present. Patent Owner argues that the claims are different, which is true. However, Patent Owner does not demonstrate any differences in the scope of what the claims cover, yet alone any significant differences. Patent Owner also does not dispute that claims need not be identical for issue preclusion to apply. Neither does Patent Owner cite any authority contradicting *Apple* or *Ohio Willow Wood*. Rather, Patent Owner argues the issue is different based on a functional difference based on how Sinharoy functions. PO Reply Br. 3–4.

We therefore determine that the Sinharoy grounds are shown to be estopped by issue preclusion because the '1344 IPR determined the substantially the same claims there unpatentable over Sinharoy.

D. Obviousness Grounds

A claim is unpatentable as obvious under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said

subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations, including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called “secondary considerations,” including commercial success, long-felt but unsolved needs, failure of others, and unexpected results. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). In this proceeding, neither party has presented evidence directed to the fourth *Graham* factor.

E. Claims 1, 2, 4–7, and 11–14 as Obvious Over Agarwal (Ground 1)

Petitioner asserts that Agarwal teaches or suggests each limitation of claims 1, 2, 4–7, and 11–14 and provides a limitation-by-limitation analysis. Pet. 6–37; Ex. 1003 ¶¶ 113–288.

1. Independent Claim 1

a. Preamble 1pre: “An apparatus comprising.”

Petitioner contends Agarwal teaches the preamble, if limiting. Pet. 10 (citing Ex. 1003 ¶¶ 122–130). For example, Petitioner points to Agarwal Figure 2 as illustrating an “exemplary multicore processor 200 having a tiled processor architecture.” *Id.* at 10 n.3 (citing Ex. 1005, 11:53–54). Patent Owner does not dispute Petitioner’s assertion.

We determine that Petitioner demonstrates that the preamble of claim 1 is taught or suggested by Agarwal.

b. Limitation 1a: “a multicore processor including a plurality of cores.”

Petitioner relies on Agarwal to meet this claim limitation. Pet. 11–12 (citing Ex. 1003 ¶¶ 122–130). Petitioner cites to “Agarwal’s Figure 6B shows a multicore processor with multiple tiles, each including a processor (i.e., ‘plurality of cores’).” *Id.* (citing Ex. 1005, 20:1–4, Fig. 6B (annotated); Ex. 1003 ¶¶ 128–130). Patent Owner does not dispute Petitioner’s assertion.

We determine that Petitioner demonstrates that this limitation of claim 1 is taught or suggested by Agarwal.

c. Limitation 1b: “a shared cache memory”

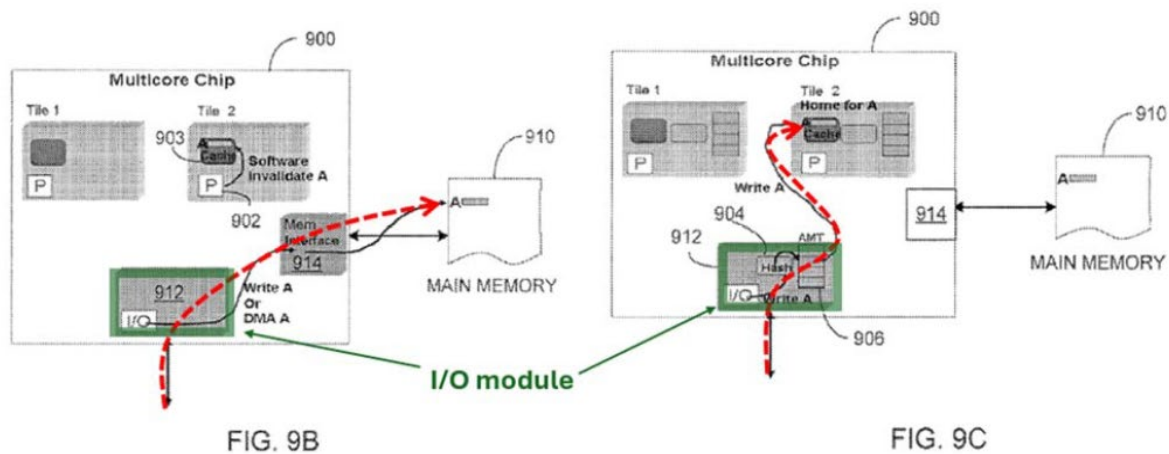
Petitioner asserts that Agarwal meets this limitation. Pet. 12–13 (citing Ex. 1003 ¶¶ 131–142). Petitioner cites Agarwal’s Figure 6B again as showing “each tile includes an L2/L3 cache 602 whose L3 entries can be shared by all tiles’ cores (i.e., ‘a shared cache memory’).” *Id.* (citing Ex. 1005, 20:1–16, 23:45–51, Fig. 6B (annotated); Ex. 1003 ¶¶ 138–141). Patent Owner does not dispute Petitioner’s assertion.

We determine that Petitioner demonstrates that this limitation of claim 1 is taught or suggested by Agarwal.

d. Limitation 1c: “an integrated input/output (IIO) module to interface between the multicore processor and at least one IO device coupled to the multicore processor”

Petitioner asserts that Agarwal meets this limitation. Pet. 13–15 (citing Ex. 1003 ¶¶ 143–162). Petitioner argues Agarwal’s multicore processor ([1a]) includes an “I/O module.” *Id.* at 13 (citing Ex. 1005, 10:31–32, Figs. 9A–9C; Ex. 1003 ¶ 144). Petitioner argues Agarwal’s

“Figure 9B shows, multicore processor 900 includes an ‘I/O module 912’ integrated onto the same multicore processor as the plurality of processor cores.” *Id.* at 14 (citing Ex. 1005, 28:11–22, 29:5–7, Fig. 9B (annotated); Ex. 1003 ¶¶ 150–152). Annotated Figures 9B and 9C are reproduced below.



Figures 9B and 9C annotated are diagrams of a multicore chip with an I/O module.

Ex. 1005, 10:31–32; Pet. 15 (Petitioner’s Figs. 9B and 9C annotations). Petitioner argues Agarwal’s “I/O module 912 supports transactions that send data from an off-chip IO device (1) directly to ‘main memory’ (Fig. 9B and 28:11–22, 28:33–43) or (2) directly into the L2/L3 cache (Fig. 9C and 28:23–25, 29:5–7).” Pet. 15 (also citing Ex. 1003 ¶¶ 158–161 (additional support from the Hassoun Declaration)). Patent Owner does not dispute Petitioner’s assertion.

We find that Petitioner demonstrates that this limitation of claim 1 is taught or suggested by Agarwal.

e. Limitation 1d: “a caching agent to perform cache coherency operations for the plurality of cores and the IIO module”

Petitioner asserts that Agarwal meets this limitation. Pet. 15 (citing Ex. 1003 ¶¶ 163–193). Petitioner alleges that:

To manage cache coherency, *Agarwal* teaches several cache coherence management techniques, including Shim Cache Coherence (SCC) and Proxy Cache Coherence (PCC) protocols. (*Agarwal*, 16:26-20:34, 27:39-30:4.) Both are directory-based protocols used to maintain cache coherence (*Agarwal*, 1:46-50, 2:21-24). Both protocols employ: (1) a directory controller and (2) directory storage that collectively operate as the claimed “caching agent” to perform cache coherency operations for Agarwal’s plurality of cores and IIO module. (*Agarwal*, 2:47–54, 8:38–42, 1:46–50, 11:12–26, 12:28–32, 27:39–28:9) (Ex. 1003 ¶¶ 166–174.)

Id. at 15–16. Petitioner further alleges that “[a] particular cache operation is managed by one of the tiles in the multiprocessor, referenced as the ‘home tile’ for that cache memory operation.” *Id.* at 16 (citing Ex. 1005, 16:26–45, 23:51–53.)

Petitioner argues “*Agarwal* depicts a PCC tile architecture configured in Figures 6A–6B.^[9] Each tile includes a processor 600, L2/L3 cache 602, directory controller 603, and directory storage 604.” Pet. 17–18 (citing Ex. 1005, Fig. 6B; Ex. 1003 ¶¶ 184–186).

Petitioner relies in part on Figure 9A annotated, reproduced below.

⁹ As noted above, Figures 6A and 6B are very similar. *See supra* Section I.H.1 (reproducing Figure 6B).

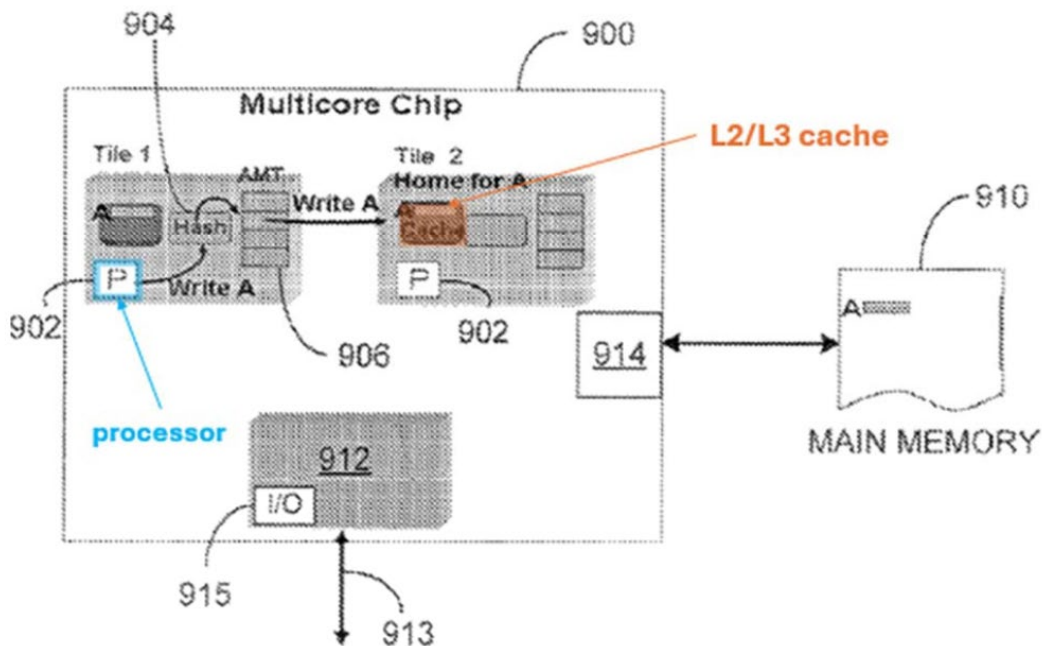


FIG. 9A

Figure 9A shows a multicore processor 900 in which a processor 902 in Tile 1 is writing a shared cache location having address A.

Ex. 1005, 27:47–48; Pet. 19 (Petitioner’s annotations). Petitioner alleges that Agarwal describes “cache coherence operations for the plurality of cores.” Pet. 19 (citing Ex. 1005, Fig. 9A; Ex. 1003 ¶¶ 187–188).

Furthermore, Petitioner alleges that “[w]hen Tile 1’s processor 902 writes data to memory location having address A, Tile 1 sends the write request to the home tile of address A (Tile 2) based on a hash function and an address mapping table (AMT) or TLB.^[10]” *Id.* at 19–20 (citing Ex. 1005, 27:46–

¹⁰ “Translation Lookaside Buffer.” Ex. 1009 (Keckler et al., “*Multicore Processors and Systems*,” (Copyright 2009, Springer Science+Business Media, LLC)), 195 (Petitioner’s pagination).

28:8, 29:17–30:11, 23:6–53, 27:32–38, 17:42–67, 19:3–29). Petitioner further alleges that Agarwal teaches “[c]oherence processing at the home tile for memory access requests from the I/O module 912 (‘I/O coherence’) includes a set of actions that are similar to those for PCC.” *Id.* at 20 (citing Ex. 1005, 29:29–31; Ex. 1003 ¶¶ 190–193). Patent Owner does not dispute Petitioner’s assertion.

We find that Petitioner demonstrates that this limitation of claim 1 is taught or suggested by Agarwal.

f. Limitation 1e¹¹: “the caching agent a single caching agent for the multicore processor and including a plurality of distributed portions each associated with a corresponding one of the plurality of cores”

Petitioner asserts that Agarwal meets this limitation. Pet. 22 (citing Hassoun Dec. ¶¶ 195–203). Citing its showing for limitation 1d above, Petitioner alleges “Agarwal discloses a PCC protocol with distributed L3 caches and cache management in each tile’s directory controller and directory storage.” *Id.* at 22–23 (citing Ex. 1005, 16:26–20:45, 27:39–30:4, Figs. 6A–6B; Ex. 1003 ¶¶ 196–200).

Petitioner alleges the caching agent portions are “each associated with a corresponding one of the plurality of cores.” Pet. 23. The preceding is

¹¹ Petitioner notes that “[t]he language ‘the caching agent a single caching agent’ lacks a verb. Pet. 22 n.6; *see also* ’1344 IPR Final Decision; ’1344 IPR Ex. 1001, 7:58–59 (claim 1 of ’895 patent reciting “the caching agent *is* a single caching agent”). Neither party argues any issue based on the missing verb. We do not find the missing verb affects our analysis and proceed without need for further discussion.

based on the '228 patent's teaching that "the caching agent can be associated with a corresponding core *and* LLC bank or slice." *Id.* (quoting Ex. 1001, 2:67–3:3). Similarly, according to Petitioner, Agarwal teaches "[a]t least some of the directory controllers are associated with a processor core." *Id.* (citing Ex. 1005, 3:42–44, 30:65–67 (claim 9)).

Patent Owner disputes limitation 1e has been shown. We take Patent Owner's arguments in the order presented in Patent Owner's Response followed by our analysis.

Patent Owner first argues Agarwal teaches a "distributed directory" and is not a "single caching agent." PO Resp. 9–10 (citing Bagherzadeh Dec. (Ex. 2006) ¶¶ 35–37). Patent Owner acknowledges that Agarwal's "home tiles" include "a directory controller responsible for managing cache coherence for a specific region of memory." *Id.* at 9 (citing Bagherzadeh Dec. ¶ 35 (citing Ex. 1005, 3:7–14, 3:21–24, 3:39–41, 3:55–61, 4:9–12, 4:40–45, 4:55–57, 17:42–57, 23:51–53)). However, Patent Owner alleges "[e]ach directory controller acts independently for its assigned memory region." *Id.* at 9–10 (citing Ex. 2006 ¶ 36). Thus, "Agarwal's architecture is characterized by a set of distributed and independent directory controllers, each managing coherence for a distinct memory region." *Id.* at 10 (citing Bagherzadeh Dec. ¶ 38). Patent Owner concludes Agarwal discloses a "distributed directory protocol, where each directory controller is responsible for its own partition of memory, and there is no central or logically unified agent managing coherency for the entire multicore processor." *Id.* (citing Bagherzadeh Dec. ¶ 41; Ex. 1005, 16:26–45, 23:51–53).

Based on the preceding, Patent Owner concludes that Agarwal does not disclose a “single caching agent” as required by limitation 1e. PO Resp. 10–11 (citing Bagherzadeh Dec. ¶ 39; Ex. 1001, 2:66–3:3, 6:22–29). Patent Owner explains the importance of this alleged distinction by arguing the advantage the ’228 patent describes and claims is avoiding the “need for multiple caching agents.” *Id.* at 11 (citing Bagherzadeh Dec. ¶ 40).

Patent Owner argues the Hassoun Declaration testimony does not show a “single caching agent” is taught by Agarwal. PO Resp. 11 (citing Ex. 2006 ¶ 64). The cited portion of the Hassoun Declaration asserts Agarwal teaches

a distributed caching agent in which individual portions of the caching agent collectively handle the different portions of a shared cache to form a collective single caching agent that handles both CPU and IO module cache coherency for multicore architectures with an integrated IO module.

Ex. 1003 ¶ 64. Patent Owner argues “collective action” is not the same as a “single caching agent.” PO Resp. 11–12. Additionally, Patent Owner argues Agarwal does not teach “a plurality of distributed portions each associated with a corresponding one of the plurality of cores.” *Id.* at 12–14. Specifically, “Agarwal’s directory controllers serve—are associated with—memory regions, not cores.” *Id.* at 13 (citing Ex. 1005, 19:7-29).

Patent Owner argues that Agarwal does not show a “single caching agent that manages coherency for both the cores and the I/O module; rather, the I/O module will use whichever directory controller manages the target memory region.” PO Resp. 14–15 (citing Ex. 1005, 29:5–31; Ex. 2006

¶¶ 29–51; Ex. 2008,¹² 67:11–15). Relying on the Bagherzadeh Declaration, Patent Owner argues “[t]here is no single agent responsible for coherency for all cores and the I/O module together.” *Id.* at 15 (citing Ex. 2006 ¶ 51).

Responding to the argument that no single caching agent is responsible for coherency for all cores, Petitioner asserts that managing “individual ‘slices’ or ‘partitions’ or ‘banks’ (in the [’]228 patent’s words) which are memory regions” is the same as managing the overall shared cache. Pet. Reply 2. Petitioner cites to the claim language of the ’228 patent, which recites a “‘shared cache’ among cores for which ‘cache coherency operations’ are performed by the distributed caching agent portions.” *Id.* (citing Ex. 1001, 7:53–56 (claim 4¹³)).

Petitioner also cites the ’228 patent as disclosing that each core has a caching agent. Pet. Reply 4.

[P]rocessor 700 includes a distributed configuration having partitions or slices each including a core 710 and a partition of a caching agent 715 and a LLC 720. Note that while distributed caching agents are shown, understand that these distributed portions form a single caching agent, and which is configured to handle cache coherency operations both for the cores as well as an IIO module.

¹² We believe this is the wrong Exhibit and presume the correct Exhibit is the Deposition of Marwan Hassoun, Ph.D. (Ex. 2007, 67:11–15) (in reference to Agarwal, “the directory controller for that home tile handles the I/O request . . .”).

¹³ “4. The apparatus of claim 1, wherein the shared cache memory comprises a last level cache (LLC) including a plurality of distributed portions each associated with a corresponding one of the plurality of cores.”

Id. (quoting Ex. 1001, 6:22–29). Other portions of the '228 patent are also quoted by Petitioner as disclosing that portions of the caching agent are associated with “memory regions,” denominated as slices, partitions, and banks. *Id.* (quoting Ex. 1001, 3:1, 2:55).

Petitioner further argues that Figure 6 of the '228 patent “depicts the claimed ‘single caching agent’ with distributed portions, and cores and distributed slices / partitions of the shared cache / LLCs (memory regions) all associated with each other – shown as 710₀/715₀/720₀ to 710_n/715_n/720_n.” Pet. Reply 3. Petitioner’s annotation of Figure 6 is reproduced below.

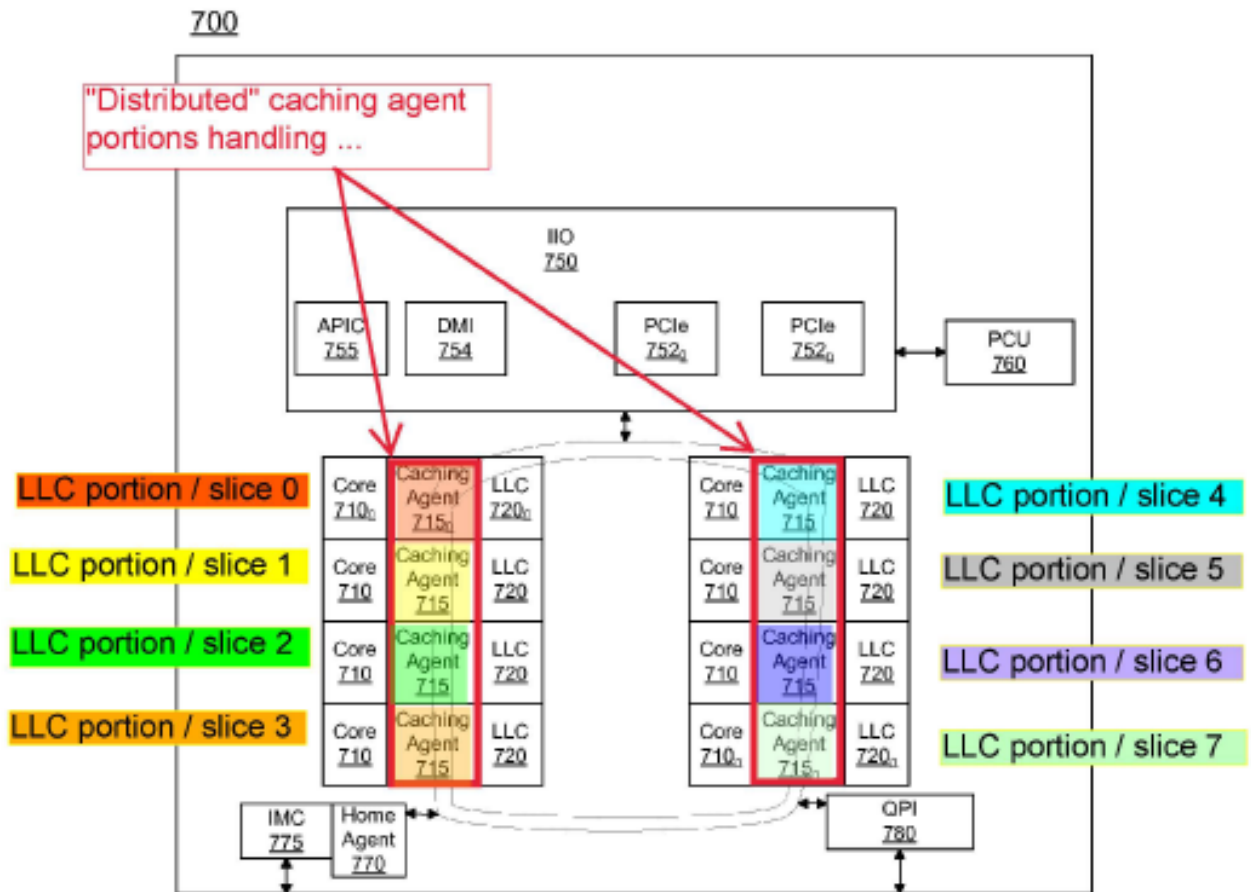


FIG. 6

Figure 6 is a block diagram of a multicore processor in an embodiment of the invention.

Ex. 1001, 6:20–22 (annotated at Pet. 47; Pet. Reply 3, 5, 13).

Petitioner argues “[the I/O module together] portions are associated with shared cache portions most naturally indicates that each caching agent portion manages coherency for its associated cache memory portion / region to achieve collective shared cache coherency.” Pet. Reply 4. According to Petitioner, Patent Owner’s position “requires that the ’228 patent exclude such distributed processing of the cache coherency operations – yet there is no passage in the ’228 indicating any such exclusion.” *Id.* at 5.

Patent Owner’s argument that distributed caching agent portions do not meet the claim language is all but identical to the one made and rejected in the ’1344 IPR in connection with Sinharoy. ’1344 IPR Final Decision, 23–28. The related arguments are persuasively rebutted in the Reply. Pet. Reply 10–13. Specifically, Petitioner argues “*Agarwal*’s distributed controllers (collectively) ‘include various coherency logic to perform cache coherency operations on behalf of the cores and IIO module, and to enable coherent access with regard to both the core transactions and IO device transactions’ as taught in the ’228 patent.” Pet. Reply 7 (citing Ex. 1001, 2:62–65; Ex. 1005, 17:42–18:24; 27:40–59; 30:5–26; Petition 16–17, 19–23); *id.* (arguing “*Agrwal*’s distributed cache directory coherence protocols such as the ‘PCC’ (Response at 16, 18; Petition, 7-10, 16-23) mirror protocols used by the ’228 patent.” (citing Ex. 1001, 4:1–4, 3:62–63)). Further, as Petitioner argues, Agarwal specifically teaches that “[a]t least some of the directory controllers are associated with a processor core . . .” *Id.* at 11 (citing Ex. 1005, 3:42–44, claim 9; Petition, 23–24).

Patent Owner’s next argument is that Petitioner has failed to show a reasonable basis or motivation to “modify Agarwal’s distributed architecture to be a logically unified single caching agent with distributed portions, as recited in Claim 1.” PO Resp. 15–17 (citing Bagherzadeh Dec. ¶¶ 52–57). Several reasons are advanced in support of this contention. The first reason relied on by Patent Owner is that Agarwal’s tiles are “self-contained unit with its own local resources, including cache and directory controller.” *Id.* at 15. This argument is not persuasive because, as Petitioner argues Agarwal’s “‘proxy cache coherence’ protocol (‘PCC’) teaches the directory controller and directory storage can be distributed across ‘tiles’ (or ‘nodes’) in a tiled multicore processor.” Pet. 7 (citing Ex. 1005, 11:22–26¹⁴; Hassoun Dec. ¶¶ 68–69). Accordingly, Agarwal’s tiles are not necessarily “self-contained” in that Agarwal teaches that the tiles can be “associated with” the directory. Ex. 1005, 11:22–26.

Patent Owner argues there is no association between the cores and the distributed portions. PO Resp. 12–13. This is not persuasive because Agarwal teaches that the tiles can be “associated with” the directory. Ex. 1005, 11:22–26. Directories in turn “maintain cache coherence.” *Id.* at 1:46–47; *see also id.* at 11:22–26 (“directory can be associated with that node’s higher level cache”).

Patent Owner’s remaining arguments regarding combining the different teachings of Agarwal are related to Patent Owner’s “single caching

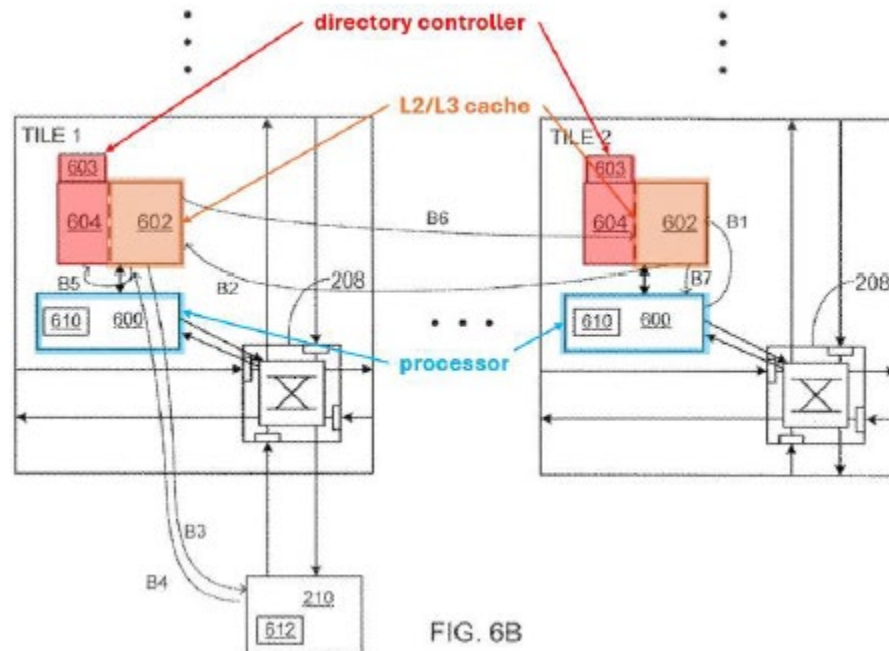
¹⁴ “(2) A directory can be located on a node (e.g., a core or tile) in the multicore system. The directory can be associated with that node's higher level cache or the directory can be standalone by itself. This second scheme is called proxy cache coherence or PCC.”

agent” arguments. PO Resp. 15–20. That is the overriding issue which we have found is taught or suggested by Agarwal. We note that “centralized control” argument is not part of claim 1. *See* Pet. Reply 6 (citing PO Resp. 12, 18). Even assuming “centralized control” is at issue, the collective distributed protocols of Agarwal manage the shared cache, i.e., common or “centralized control.” *Id.* at 9.

Patent Owner argues sharing cache lines does not teach logical unification of caching agents. PO Resp. 17. “In other words, each directory controller still acts autonomously for its assigned memory region.” *Id.* at 18 (citing Bagherzadeh Dec. ¶ 60 (citing Ex. 1005, 19:7–20:25)). The Bagherzadeh Declaration explains autonomously as meaning “each controller acts independently for its own allocated memory region.” *Id.* According to Dr. Bagherzadeh, this is fundamentally different from the claimed “single caching agent” which is allegedly “a unified logic and global view.” *Id.* (citing Ex. 1001, 2:36–65).

The eighty-five lines of Agarwal cited in the Bagherzadeh Declaration (Ex. 1005, 19:7–20:25) explain the “steps for maintaining cache coherence” in Figures 6A and 6B. Ex. 1005, 19:4–6. Dr. Bagherzadeh cites this block of text but fails to explain how this block of text supports his opinion. Because his testimony is conclusory and lacks sufficient explanation, we find his testimony on this point to be entitled to little, if any weight. This is in contrast to the Petition which explains Figure 6B of Agarwal, citing the same section but pointing to fifteen lines of the disclosure. Pet. 9 (citing Ex. 1005, Fig. 6B, 20:1–16; Hassoun Dec. ¶¶ 78–81). The Hassoun Declaration at paragraph 78 includes an annotated Figure 6B, stating “each tile includes an L1 cache 610 within the processor 600 and an L2/L3 cache

602 that includes not only private L2 entries but also L3 entries that can be shared by all cores in the tiles.” Hassoun Dec. ¶ 78. As annotated in the Hassoun Declaration, Figure 6B of Agarwal is reproduced below.



Agarwal Annotated Figure 6B.

Ex. 1003 ¶ 78. Because Dr. Hassoun’s testimony is consistent with the disclosure of Agrawal, we find it is entitled to weight and we give credit to his testimony over that of Dr. Bagherzadeh.

Patent Owner alleges that “protocol cooperation” between controllers does not teach “single caching agent.” PO Resp. 18–20. We are not persuaded by these arguments. Under the PCC protocol

the tile that serves as a home location for a given cache line is called the proxy tile. Because the proxy tile serves as the proxy for the directory memory, the scheme is called proxy cache coherence or PCC.

Ex. 1005, 16:26–20:45, 27:39–30:4, Figs. 6A–6B.

Patent Owner argues this protocol cooperation does not mean that Agarwal teaches a single caching agent. PO Resp. 18 (citing Hassoun Dec. ¶ 64). Patent Owner alleges this is not “centralized control; i.e., not a logical unity of directory controllers.” *Id.* at 18–19 (citing Bagherzadeh Dec. ¶¶ 65–70). This argument is unavailing because, as previously discussed, the claims do not recite “centralized control.” Neither does Patent Owner advance any construction or claim interpretation argument that would require “centralized control.”

We are not persuaded by Dr. Bagherzadeh’s conclusory testimony that “Agarwal’s coherence protocol ensures system-wide consistency of memory” is different from “logical unity of controllers.” Bagherzadeh Dec. ¶ 65. We again give credit to the Hassoun Declaration testimony that Agarwal is like the ’228 patent solution and teaches “a distributed caching agent in which individual portions of the caching agent collectively handle the different portions of a shared cache to form a collective single caching agent that handles both CPU and IO module cache coherency for multicore architectures with an integrated IO module.” Hassoun Dec. ¶¶ 64–69 (citing Ex. 1005 Abs., 1:34–53, 11:12–26).

What Patent Owner states that corresponds to Petitioner’s allegation in Petitioner’s Reply is that Agarwal teaches “a set of distributed and independent directory controllers, each managing coherence for a distinct memory region.” PO Resp. 10 (citing Bagherzadeh Dec. ¶ 38 (“Agarwal’s architecture is characterized by a set of distributed and independent directory controllers, each managing coherence for a distinct memory region as the ‘home tile,’ and handling both processor and IO module transactions for their assigned regions.”)); *see also* Pet. Reply 6 (Agarwal’s “directory

controller/storage pairs (‘distributed portion’ [of the caching agent]) manage coherency operations for portions of the overall shared cache.”).

Patent Owner’s Sur-Reply again argues the “collective action” argument of Petitioner is “inconsistent with both the claim language and the intrinsic record.” PO Sur-Reply 6. This position was previously addressed and does not change our finding that the “single caching agent” is taught or suggested by Agarwal. In another repetitive argument, Patent Owner argues that the term under consideration is “not satisfied by a collection of independent controllers that exchange messages under a distributed protocol” because they are not a “logically unified caching agent.” PO Sur-Reply 5–6 (citing Pet. Reply 7–9). Our analysis and determination on this issue remains the same. We find that Petitioner demonstrates that this limitation of claim 1 is taught or suggested by Agarwal.

2. *Dependent Claim 2*

Claim 2: “The apparatus of claim 1, wherein the caching agent includes a table of requests to store pending requests for the plurality of cores and the IIO module.”

Petitioner asserts that Agarwal meets all limitations of claim 2. Pet. 24 (citing Ex. 1003 ¶¶ 204–213). Petitioner specifically cites to Agarwal’s “buffer.” *Id.* (citing Ex. 1005, 27:21–25, 16:1–5, 20:53–54).

Patent Owner disputes this showing and makes the following arguments. PO Resp. 21–27. “The ‘table of requests’ (TOR) is described as a centralized structure within the caching agent that stores pending requests from both the plurality of cores and the IIO module and is part of the caching agent’s logic for managing and arbitrating requests.” PO Resp. 21 (citing Ex. 1001, 2:44–53, 2:66 – 3:3; Fig. 2; Ex. 2006 ¶ 76). “Claim 2

makes clear *the* caching agent (singular) includes a table of requests (singular) for all sources, which implies a unified, centralized structure within the caching agent that stores all incoming requests.” *Id.* at 22 (citing Ex. 2006 ¶ 76). “There is no disclosure in Agarwal of a centralized table or buffer that stores pending requests from both the processor cores and the I/O module” . . . or that the buffers “are shared resources.” *Id.* at 23. The argument that “Agarwal’s per-tile buffer is equivalent to the ’228 patent’s TOR ignores the requirement that the TOR is a shared structure for *all* sources, not a local buffer. There is no such shared structure in Agarwal.” *Id.* at 24.

Petitioner argues the ’228 patent’s “Figure 2 shows the caching agent’s components (including a TOR) and the Figure 2 ‘caching agent can be distributed such that each of different portions of the caching agent can be associated with a corresponding core and LLC bank or slice.’” Pet. Reply 14 (citing Ex. 1001, 2:67–3:3). Petitioner argues Figure 5 of the ’228 patent shows a “separate TOR in each core.” *Id.* at 16–17 (citing Ex. 1001, Fig. 5). Petitioner cites the ’228 patent’s disclosure that “[c]ore 420₀ may be issuing non-posted requests, such as core memory mapped IO (MMIO) read requests from a non-posted queue 422₀ to a TOR 444₀ of caching agent 440₀.” *Id.* (quoting Ex. 1001, 5:31–33). In addition, “Figure 5’s discussion confirms that the caching agent’s components are included in each distributed portion in each core. For example, the ‘queues 211–213’ are part of the caching agent in Figure 2.” *Id.* (citing Ex. 1001, 2:42–44).

Patent Owner’s Sur-Reply regarding claim 2 repeats arguments addressed above. For example, Patent Owner argues the “table of requests is a single, shared structure within the caching agent, not a collection of

independent buffers dispersed across separate controllers.” PO Sur-Reply 20 (citing Bagherzadeh Dec. ¶¶ 74–76, 79–81).

Dr. Bagherzadeh describes the TOR as follows.

The recited “table of requests” (TOR) within the caching agent, which stores pending requests for both the cores and the IIO module, is described as *a buffer* or temporary storage for holding incoming requests from both the cores and the IIO module and is part of the single caching agent’s logic for managing and arbitrating requests.

Id. ¶ 74 (citing Ex. 1001, 2:44–53, 2:66 – 3:3) (emphasis added).

Agarwal teaches a buffer, which is cited by Petitioner in its showing regarding claim 2. Pet. 24 (citing Ex. 1005, 27:21–25, 16:1–5, 20:53–54). We find Agarwal’s buffer corresponds to the TOR recited in claim 2 of the ’228 patent.

We find that Petitioner demonstrates that claim 2 is taught or suggested by Agarwal.

3. Dependent Claim 4

Claim 4: “The apparatus of claim 1, wherein the shared cache memory comprises a last level cache (LLC) including a plurality of distributed portions each associated with a corresponding one of the plurality of cores.”

Petitioner asserts that Agarwal meets all limitations of claim 4. Pet. 25–27 (citing Ex. 1003 ¶¶ 204–213). Petitioner argues “*Agarwal* discloses an L2/L3 cache 602 for each tile in the multicore processor with the L3 cache lines being shared among all the tiles (i.e., ‘a last level cache (LLC)’).” *Id.* at 25–26 (citing showing for limitation 1b above; Ex. 1005, 20:1–16, Fig. 6B). Patent Owner does not argue that claim 4 is patentable over Agarwal. See generally PO Resp.

We find that Petitioner demonstrates that claim 4 is taught or suggested by Agarwal.

4. *Dependent Claim 5*

Claim 5: “The apparatus of claim 4, wherein the plurality of distributed portions of the caching agent are coupled via a ring interconnect.”

Petitioner asserts that Agarwal meets all limitations of claim 5. Pet. 27–28 (citing Ex. 1003 ¶¶ 223–226). Petitioner argues one disclosed mesh network of Agarwal’s tiles “are interconnected over a network that includes . . . a ring” *Id.* at 28 (citing Ex. 1005, 3:45–47, 7:16–18, 11:66–12:2, 31:1–3 (claim 10)). Patent Owner does not argue that claim 5 is patentable over Agarwal. *See generally* PO Resp.

We find that Petitioner demonstrates sufficiently that claim 5 is taught or suggested by Agarwal.

5. *Dependent Claim 6*

Claim 6: “The apparatus of claim 1, wherein the caching agent is to receive an allocation transaction from the IIO module, the allocation transaction received in the IIO module from an IO device coupled to the multicore processor.”

Petitioner asserts that Agarwal meets all limitations of claim 6. Pet. 28–30 (citing Ex. 1003 ¶¶ 227–235). Petitioner argues Agarwal discloses an allocation transaction in disclosing “[a]n I/O module is also able to . . . store data directly into the cache of a given tile” *Id.* at 29 (citing Ex. 1005, 28:23–32, Ex. 1003 ¶¶ 229–234); *id.* at 25–26 (citing showing for limitation 1b above; Ex. 1005, 20:1–16, Fig. 6B); *see also* Ex. 1001, 3:47–51 (“[T]his allocation transaction can be a request to write data directly into

a cache memory, . . . of the processor, and it can be received from an IO device.”).

Without construing “allocation transaction,” Patent Owner argues that an “allocation transaction . . . in the context of the ’228 patent, is a transaction initiated by an IO device (via the IIO module) that causes data to be directly stored (allocated) into the shared cache memory of the multicore processor.” PO Resp. 27. “Agarwal’s ‘cache coherence case’ concerns *core-initiated* DMA [direct memory access] transactions, not transactions initiated by external IO devices.” *Id.* at 28–29 (citing Ex. 2007 (“Hassoun Deposition”), 69:18–70:20). Patent Owner does not argue that claim 4 is patentable over Agarwal. *See generally* PO Resp. Patent Owner then contends “[c]laim 6’s requirement that the allocation transaction be ‘received in the IIO module *from an IO device* coupled to the multicore processor’ is missing.” *Id.* at 29 (citing Ex. 2006 ¶ 94).

We are not persuaded. Agarwal teaches “[a]n I/O module is also able to . . . store data directly into the cache of a given tile” Pet. Reply 19 (citing Ex. 1005, 28:23–32.)

We find that Petitioner demonstrates that claim 6 is taught or suggested by Agarwal.

6. *Dependent Claim 7*

Claim 7: “The apparatus of claim 6, wherein the caching agent is to directly store data of the allocation transaction into the shared cache memory and to not initiate a memory transaction to a system memory coupled to the multicore processor.”

Petitioner asserts that Agarwal meets all limitations of claim 7. Pet. 30–32 (citing Ex. 1003 ¶¶ 236–247). Petitioner argues Agarwal

discloses a “cache coherence case” the “I/O write data is written directly into the shared L3 cache of the appropriate home tile (e.g., Tile 2 in Fig. 9C).”

Id. at 30 (citing showing for claim 6; Ex. 1005, 28:23–32, 29:5–30:11).

Patent Owner does not argue that claim 7 is patentable over Agarwal. *See* generally PO Resp.

We find that Petitioner demonstrates that claim 7 is taught or suggested by Agarwal.

7. Independent Claim 11

Independent claim 11 is similar to independent claim 1. Petitioner asserts that Agarwal meets all limitations of claim 11. Pet. 32 (citing Ex. 1003 ¶¶ 256–261). Petitioner’s analysis of claim 11 incorporates much of its analysis of claims 6 and 7. *Id.*

Patent Owner relies on arguments made in connection with claim 6 re “allocation transaction.” PO Resp. 29–30 (citing Bagherzadeh Dec. ¶ 97). Patent Owner does argue that in “Agarwal, an I/O module delivers a transaction to whichever home tile chosen by a hash of the associated address and the directory controller on that tile handles coherence operations.” *Id.* at 31 (citing Bagherzadeh Dec. ¶ 100). The ’228 patent differs in that it “teaches proxying IIO traffic a shared CPU/IIO coherency agent, with the IIO explicitly forwarding the allocation transaction to that shared agent. That proxying step (and the identity of the recipient) matters for^[15] and is missing from Agarwal. *Id.* (citing Bagherzadeh Dec. ¶¶ 101–102). Petitioner argues, and we agree, that “the specification mirrors the

¹⁵ PO Resp. 29 (citing ’228 patent, 8:18–21, limitation 11b).

claim language at Ex-1001, 3:51-56, but it does not impose further restrictions (proxy or otherwise) on the forwarding, contrary to PO's position." Pet. Reply 21 (citing Ex. 1001, 3:51–56 (explaining Figure 3)). Patent Owner's Sur-Reply does not address claim 11 any further. *See generally* PO Sur-Reply.

We find that Petitioner demonstrates that this limitation of claim 11 is taught or suggested by Agarwal.

8. *Dependent Claim 12*

Claim 12: "The method of claim 11, further comprising accessing the data in the shared cache memory via at least one of the plurality of cores of the multicore processor a number of times, without sending a transaction to a system memory coupled to the multicore processor."

Petitioner asserts that Agarwal meets all limitations of claim 12. Pet. 34 (citing Ex. 1003 ¶¶ 263–274). Petitioner's analysis of claims 7 and 11 is cited. *Id.* at 34. Patent Owner does not argue that claim 7 is patentable over Agarwal. *See generally* PO Resp.

We find that Petitioner demonstrates that this limitation of claim 12 is taught or suggested by Agarwal.

8. *Dependent Claim 13*

Claim 13: "The method of claim 12, further comprising sending data of the cache line to the system memory from the shared cache memory if the cache line is evicted from the shared cache memory, when the cache line includes modified data."

Petitioner alleges Agarwal discloses claim 13, because a person of ordinary skill in the art "would understand this claim to be, at a minimum, obvious because it describes typical cache operation in which evicted cache

lines with modified data are written to memory to preserve the modifications.” Pet. 36 (citing Ex. 1003 ¶¶ 275–285). Patent Owner does not argue that claim 13 is patentable over Agarwal. *See generally* PO Resp.

We find that Petitioner demonstrates that this limitation of claim 13 is taught or suggested by Agarwal.

9. *Dependent Claim 14*

Claim 14: “The method of claim 12, further comprising directly storing the data of the allocation transaction into the shared cache memory without communicating with the system memory.”

Petitioner asserts that Agarwal meets all limitations of claim 14. Pet. 37 (citing Hassoun Dec. ¶¶ 286–288). Petitioner relies on its analysis of claims 7, 11, and 12. *Id.* (citing Ex. 1005, 23:66–41, 28:23–32, 29:5–30:11, Fig. 9C).

Patent Owner alleges Petitioner “does not identify any explicit teaching in Agarwal that write occurs ‘without communicating with the system memory.’” PO Resp. 32–33 (citing Ex. 1005, 28:23–32; 29:5–30:11; Hassoun Dec. ¶¶ 286–288). Patent Owner argues “writ[ing] data directly into the L2/L3 cache” does not teach “communicating with the system memory.” *Id.* at 33–34. Patent Owner concludes by arguing the “cache-injection alternative does not, without more, establish that the cache-injection path is implemented ‘without communicating with the system memory.’” *Id.* at 34 (citing Bagherzadeh Dec. ¶ 107).

Petitioner argues that Agarwal teaches two scenarios (cases) relevant to this showing: The first case is “memory coherence” (Fig. 9B) and the second case is “cache coherence” (Fig. 9C). Pet. Reply 22. Relying on the

“cache coherence case,” Petitioner argues Agarwal teaches “the directory controller and directory storage do not initiate a memory transaction to a main memory (i.e., ‘system memory’), unlike the system memory transaction shown in Fig. 9B.” *Id.* (citing Ex. 1005, 29:5–30:11, Fig. 9C). Petitioner concludes that the “directory controller and directory storage do not initiate a memory transaction to a main memory (i.e., ‘system memory’)” unlike the “memory coherence” case of Figure 9B. *Id.* Petitioner concludes by citing Agarwal’s express teaching that if previously stored data is not needed there is no need to fetch background data. *Id.* at 23 (citing Ex. 1005, 30:12–26). Patent Owner does not respond to this argument, which we find persuasive. *See generally*, PO Sur-Reply.

We find that Petitioner demonstrates that claim 14 is taught or suggested by Agarwal.

Conclusion Ground 1

Petitioner has shown by a preponderance of the evidence that claims 1–2, 4–7, and 11–14 would have been obvious over Agarwal.

F. Claims 1, 4, 6, 7, and 11–14 as Obvious Over Sinharoy¹⁶ (Ground 2A)

Petitioner asserts that claims 1, 4, 6, 7, and 11–14 would have been obvious over Sinharoy. Pet. 37–62. Patent Owner disputes the showing of claims 1, 4, 6, and 11. *See* PO Resp. 35–42.

¹⁶ As discussed in Section II.C above, Ground 2A has been determined based on issue preclusion. Ground 2A is also determined on the merits.

1. Independent Claim 1

Petitioner asserts that Sinharoy teaches or suggests each limitation of claims 1, 2, 4–7, and 11–14 and provides a limitation-by-limitation analysis. Pet. 37–62; Hassoun Dec. ¶¶ 289–459. Only limitation 1e is disputed by Patent Owner. PO Resp. 35–37.

Petitioner’s showing for limitation 1e, “a single caching agent . . . including . . . distributed portions,” is that Sinharoy’s has the same architecture as the ’228 patent’s architecture where “each core includes: (1) a core, (2) a portion of the shared L3 cache; and (3) an L3 cache “controller.” Pet. 48 (citing Ex. 1006, 17–19). Petitioner then argues “*Sinharoy* operates just like the ’228 patent in reducing snooping, for example, by the I/O component.” *Id.* at 51 (citing Ex. 1001, 1:19–30, Hassoun Dec. ¶¶ 381–383). Petitioner emphasizes the same operations between Sinharoy as the ’228 patent by contending “the IIO module, the cache appears as a single entity and the IIO module does not need a separate I/O agent for each core – which is the problem identified in the ’228 patent’s Background.” *Id.* at 51–52 (citing Ex. 1001, 12–30; Hassoun Dec. ¶ 384). Petitioner’s argument is supported by a comparison of annotations of Figure 6 of the ’228 patent and Figure 14 of Sinharoy, reproduced below.

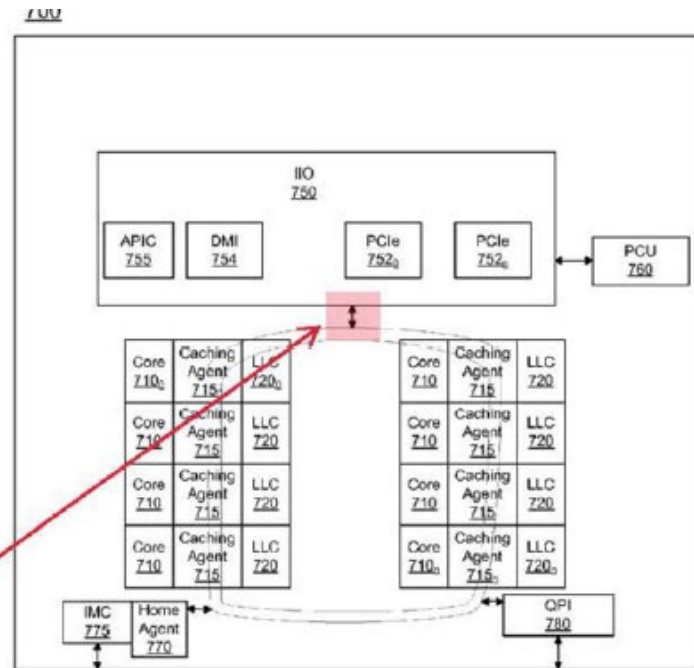
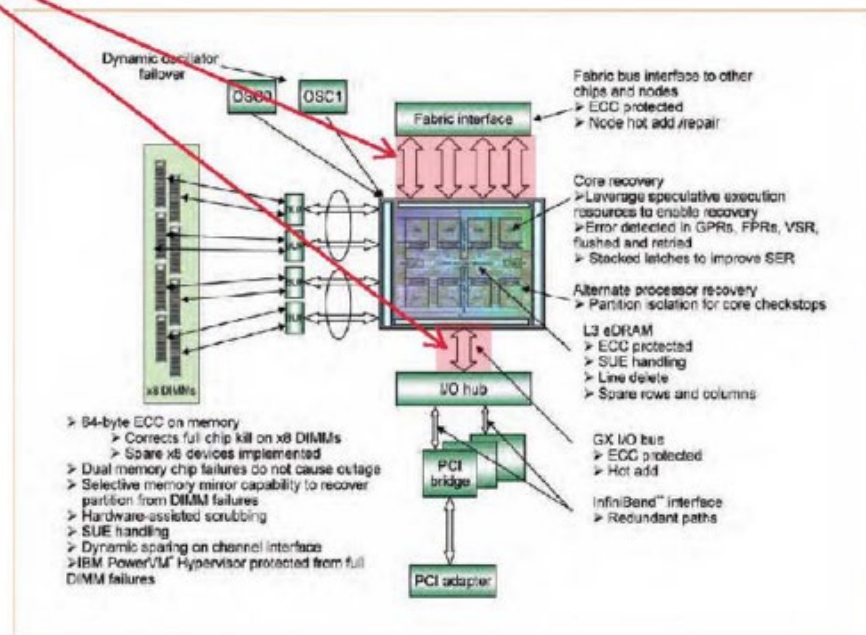
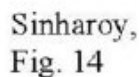
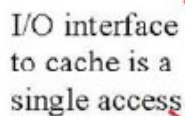


FIG. 6



Annotated Figure 6 of '228 patent and Figure 14 of Sinharoy.

Pet. 52. Petitioner compares the two figures as follows:

Sinharoy's I/O units (the two I/O controllers at bottom and the SMP interconnect at top) treat the cache as a single integrated unit—needing to only submit a single cache access (read/write) to

the cache coherency logic. *Sinharoy*'s distributed L3 caching agent logic manages that access. For example, *Sinharoy*'s L3 cache may need to perform lateral cast-out operations or other coherency logic operations, but those operations are abstracted from the I/O unit. As with the '228 patent, *Sinharoy*'s I/O devices do not need separate I/O agents to access cache associated with each core. In this manner, *Sinharoy* operates in the same manner as the [']228 patent identifies for a single caching agent. (Ex-1001, 1:12-30; Ex-1003 ¶384).

Id. at 53.

Patent Owner makes the following arguments. First, *Sinharoy* “employs multiple independent controllers (for L2, L3, and I/O), each acting as a peer on a broadcast coherence fabric, with no single agent managing coherency for both CPU and I/O domains.” PO Resp. 36 (citing Ex. 1006, 1–2, 16–21, 22–23; Bagherzadeh Dec. ¶¶ 109, 114). As such, the “individual controllers of *Sinharoy*'s L3 cache regions are not described as distributed portions” and “each L3 region controller operate independently.” *Id.* (citing Ex. 1006, 19, 23; Bagherzadeh Dec. ¶ 114). The I/O controller also participates as a peer and is not subordinate to a CPU caching agent. *Id.* at 36–37 (citing Ex. 2006 ¶¶ 114–117).

Petitioner argues that *Sinharoy* does disclose a single caching agent and asserts Patent Owner does not respond to this position. Pet. Reply 24 (citing Pet. 2–4 (citations omitted)). Patent Owner responds that *Sinharoy* “does not consolidate those agents into a single caching agent.” PO Sur-Reply 11 (citing Bagherzadeh Dec. ¶¶ 109–111, 115–117).

We find a preponderance of the evidence supports Petitioner's contention that *Sinharoy* discloses a single caching agent. Ex. 1003 ¶¶ 381–383; Ex. 1006, 17–19, Fig. 14. “[T]he '228 patent teaches at least one

embodiment of a ‘distributed’ single caching agent by having separate distributed portions associated with a slice / portion of LLC that collectively handle the cache coherency operations for the entire multi core and IIO module system.” Pet. 4. The collective handling of cache coherency is shown in Figure 6 of the ’228 patent, reproduced in an annotated version in Section II.D.1.e above.

We find that Petitioner demonstrates that claim 1 is taught or suggested by Sinharoy.

2. Dependent Claim 4

Claim 4: “The apparatus of claim 1, wherein the shared cache memory comprises a last level cache (LLC) including a plurality of distributed portions each associated with a corresponding one of the plurality of cores.”

Petitioner alleges Sinharoy discloses claim 4. Pet. 57 (citing Hassoun Dec. ¶¶ 408–416). Petitioner alleges “[t]he claimed ‘last level cache (LLC)’ refers to the highest level cache shared by all cores. ([Hassoun] ¶ 409). Sinharoy’s share L3 cache is ‘a last level cache (LLC),’ as claimed as detailed in [1d], [1e].” *Id.* (citing Ex. 1006, 17, 19, Fig. 1, Tbl. 1; Hassoun Dec. ¶¶ 410–415).

Patent Owner argues the POWER7 L3 cache of Sinharoy is not tied to any particular cache and is not “associated” with any core. PO Resp. 38–39 (citing Ex. 1006, 19–20; Bagherzadeh Dec. ¶¶ 120–122). Patent Owner argues that Sinharoy discloses a “shared” versus a “per-core” association. *Id.* at 39 (citing Bagherzadeh Dec. ¶ 122).

We find substantial evidence supports our determination that Sinharoy discloses claim 4. Claim 4 does not require “each LLC portion to be

‘accessible’ by only a single core.” *See* Pet. Reply 25. As discussed above, Figure 6 of the ’228 patent “shows the ‘distributed’ caching agent recited in the Challenged Claims and depicts separate portions of LLC associated with each of those caching agent portions.” *Id.* at 25–26 (citing Ex. 1001, Fig. 6, 6:20–39). The ’228 patent expressly discloses that “distributed portions form a single caching agent, and which is configured to handle cache coherency operations both for the cores as well as an IIO module 750.” Ex. 1001, 6:26–29.

We find that Petitioner demonstrates that claim 1 is taught or suggested by Sinharoy.

3. *Dependent Claim 6*

Claim 6: “The apparatus of claim 1, wherein the caching agent is to receive an allocation transaction from the IIO module, the allocation transaction received in the IIO module from an IO device coupled to the multicore processor.”

Petitioner alleges Sinharoy discloses claim 6. Pet. 57–58; Pet. Reply 26–27 (citing Hassoun Dec. ¶¶ 417–429). Patent Owner disputes that Sinharoy teaches “receive an allocation transaction from the IIO module.” PO Resp. 40–41.

Patent Owner asserts “[i]n Sinharoy’s ‘cache injection’ DMA traffic is directed to the correct physical L3 slice based on an address-homing function. PO Resp. 40 (citing Ex. 1006, 17; Ex. 2007, 80:4 – 81:16). “The two-hop message flow, I/O Device → IIO → Caching Agent, recited in Claim 6 is missing. Instead, the I/O controllers implement cache injection by writing into an L3 region when that region already owns the line.” *Id.*

Petitioner argues Sinharoy’s “cache injection” is a “coherent operation as for other cache operations.” *See* Pet. Reply 26 (citing Sinharoy, 17).

“*Sinharoy*’s coherence fabric (i.e., ‘the caching agent’) receives an allocation transaction from an I/O device via an integrated I/O controller or the SMP interconnect.” *Id.* at 26–27 (citing Petition, 57–58).

Patent Owner argues Sinharoy’s “cache injection is performed by integrated controllers within the processor complex as part of DMA handling and coherence operations.” PO Sur-Reply 25 (citing Bagherzadeh Dec. ¶¶ 108–110). Further, there is no showing of “an external IO device generates a transaction of the type claimed, nor does it explain how Sinharoy’s internal cache injection mechanism corresponds to the claimed allocation transaction semantics.” *Id.*

A preponderance of the evidence supports the showing that the caching agent receives an allocation transaction from the IIO module. *See* Pet. 58 (citing Ex. 1006, 17 (“An I/O device performing a direct memory access (DMA) write operation may target the operation to the cache, instead of to memory.”)).

We find that Petitioner demonstrates that claim 6 is taught or suggested by Sinharoy.

4. *Dependent Claim 7*

Claim 7: “The apparatus of claim 6, wherein the caching agent is to directly store data of the allocation transaction into the shared cache memory and to not initiate a memory transaction to a system memory coupled to the multicore processor.”

Petitioner asserts that Sinharoy meets all limitations of claim 7. Pet. 30–32 (citing Ex. 1003 ¶¶ 430–433). Petitioner argues Sinharoy discloses a

“instead of to memory” confirms the cache injection does not initiate a memory transaction. *Id.* at 58.

Patent Owner does not dispute Petitioner’s assertion. *See generally* PO Resp.

We find that Petitioner demonstrates that claim 7 is taught or suggested by Sinharoy.

5. Independent Claim 11

Independent claim 11 is similar to independent claim 1. Petitioner asserts that Sinharoy meets all limitations of claim 11. Pet. 59–60; WF 492–493 (citing Ex. 1003 ¶¶ 435–449). Petitioner’s analysis of claim 11 incorporates much of its analysis of claims 6 and 7. *Id.*

Patent Owner disputes this showing and makes the same arguments made in connection with claim 6. PO Resp. 41–42.

For the reasons discussed above in connection with claims 6 and 7, we find that Petitioner demonstrates that claim 11 is taught or suggested by Sinharoy.

6. Dependent Claim 12

Claim 12: “The method of claim 11, further comprising accessing the data in the shared cache memory via at least one of the plurality of cores of the multicore processor a number of times, without sending a transaction to a system memory coupled to the multicore processor.”

Petitioner asserts that Agarwal meets all limitations of claim 12. Pet. 60–61; Pet. Reply 27 (citing Hassoun Dec. ¶¶ 450–454). Petitioner’s analysis of claim 6 is cited. Pet. 60; Pet. Reply 27.

We find that Petitioner demonstrates that this limitation of claim 12 is taught or suggested by Sinharoy. Patent Owner does not dispute Petitioner's assertion.

7. Dependent Claim 13

Claim 13: "The method of claim 12, further comprising sending data of the cache line to the system memory from the shared cache memory if the cache line is evicted from the shared cache memory, when the cache line includes modified data."

Petitioner alleges Sinharoy discloses claim 13. Pet. 61–62; Pet. Reply 27 (citing Hassoun Dec. ¶¶ 455–457).

We find that Petitioner demonstrates that this limitation of claim 13 is taught or suggested by Sinharoy. Patent Owner does not dispute Petitioner's assertion.

8. Dependent Claim 14

Claim 14: "The method of claim 12, further comprising directly storing the data of the allocation transaction into the shared cache memory without communicating with the system memory."

Petitioner asserts that Agarwal meets all limitations of claim 14. Pet. 62 (citing Ex. 1003 ¶¶ 458–459). Petitioner relies on its analysis of claims 11 and 12. *Id.*

We find that Petitioner demonstrates that this limitation of claim 14 is taught or suggested by Sinharoy. Patent Owner does not dispute Petitioner's assertion.

Conclusion Ground 2A

Petitioner has shown by a preponderance of the evidence that claims 1, 4, 6, 7, and 11–14 would have been obvious over Sinharoy.

G. Claims 2, 3, and 16–18 as Obvious Over Sinharoy and Blumrich (Ground 2B)

Petitioner asserts that claims 2, 3, and 16–18 would have been obvious over Sinharoy and Blumrich. Pet. 63–73; Pet. Reply 27–29. We address the showing regarding claims 2, from which claim 3 depends, and 16–18. Claims 3 and 16–18 have not been previously discussed.

After reviewing the evidence and contentions, we determine that Petitioner has shown reasons to combine Sinharoy and Blumrich. Pet. 64–65; Ex. 1003 ¶¶ 474–476 (claims 2 and 3). Patent Owner does not dispute the showing.

Claim 2 recites in pertinent part that the “caching agent includes a table of requests to store pending requests.” *See* Pet. 66. Petitioner relies in part on Sinharoy’s teaching a “caching agent” performs write operations including “cache-injections.” *Id.* (citing Ex. 1006, 19). Petitioner explains that the “write operations include incoming (‘pending’) requests to write for the IIO module.” *Id.* (citing Ex. 1003 ¶ 470).

Claim 3 depends from claim 2 and recites in part that “the table of requests includes at least one dedicated entry.” *See* Pet. 68. Petitioner points out that when Sinharoy’s “write machines perform cache injection operations, they write into L3 cache.” *Id.* Thus, such “a write transaction is a ‘posted transaction.’” *Id.* (citing Ex. 1001, 5:3–7; Ex. 1006, 17, 19; Ex. 1003 ¶ 477).

Claim 16 is an independent claim similar to claim 1, discussed above. Claim 16's recitations include "a first integrated input/output (IIO) module to interface between the first multicore processor." Pet. 70. Petitioner notes that Sinharoy describes "up to 32 POWER7 chips may be connected to form a single 256-way SMP system." *Id.* at 69 (citing Ex. 1006, 23–25, Fig. 11, the table of requests includes at least one dedicated entry 13). In addition to "multicore processor," claim 16 also recites a "cache memory," which Petitioner asserts is taught by Sinharoy. *Id.* at 70 (citing Ex. 1003 ¶¶ 483–484). Similarly, Petitioner points to Sinharoy as teaching "cache memory," "caching agent," and "cache coherency operations" as recited in claim 16. *Id.* at 70–71 (citing Ex 1003 ¶¶ 487–491).

We have reviewed Petitioner's showing regarding claims 17 and 18. Pet. 74 (citing Ex. 1003 ¶¶ 507–508). Claim 17 is argued as unpatentable for the same reasons asserted in connection with claims 6 and 11. *Id.* Claim 18 is argued as unpatentable for the reasons asserted in connection with claim 12. *Id.*

Patent Owner disputes the showing for claims 2, 3, and 16–18. PO Resp. 40–49. The arguments made are ones previously addressed and found to be lacking.

Conclusion Ground 2B

Petitioner has shown by a preponderance of the evidence that claims 2–3 and 16–18 would have been obvious over Sinharoy and Blumrich.

H. Claim 5 as Obvious Over Sinharoy and Sistla (Ground 2C)

Claim 5: "The apparatus of claim 4, wherein the plurality of distributed portions of the caching agent are coupled via a ring interconnect."

Petitioner asserts that claim 5 would have been obvious over Sinharoy and Sistla. Pet. 74–77 (citing Ex. 1003 ¶¶ 109–110, 511–517).

Patent Owner argues that “Sistla’s ‘ring’ couples clusters, not distributed portions of a caching agent.” PO Resp. 50–51 (citing Ex. 1008 ¶¶ 18–19, 30; Ex. 2006 ¶ 158; Ex. 2007, 27:11–23, 29:16–30:1). Thus, according to Patent Owner, Sistla is not a “caching agent.”

We find Petitioner’s argument that Sinharoy shows “caching agent” and the combination shows ring interconnect is supported by a preponderance of the evidence. Ex. 1003 ¶¶ 109–110, 511–517 Petitioner has shown a rationale for the combination. Pet. 75.

We find that Petitioner demonstrates that claim 5 is taught or suggested by the combination of Sinharoy and Sistla.

Conclusion Ground 2C

Petitioner has shown by a preponderance of the evidence that claim 5 would have been obvious over Sinharoy and Sistla.

I. Claim 19 as Obvious Over Sinharoy, Blumrich, and Sistla (Ground 2D)

Petitioner asserts that claim 19 would have been obvious over Sinharoy, Blumrich, and Sistla. Pet. 74–77 (citing Ex. 1003 ¶¶ 109–110).

Patent Owner disputes this showing and makes the same argument as to claim 5. PO Resp. 29–30.

For the same reasons as discussed above for claim 5, we find that Petitioner demonstrates that claim 19 is taught or suggested by the combination of Sinharoy, Sistla, and Blumrich.

Conclusion Ground 2D

Petitioner has shown by a preponderance of the evidence that claim 19 would have been obvious over Sinharoy, Sistla, and Blumrich.

III. CONCLUSION

For the reasons discussed above, Petitioner has shown by a preponderance of the evidence that claims 1–7, 11–14, and 16–19 are unpatentable.¹⁷

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1–7, 11–14, and 16–19 of the '228 patent are determined to be unpatentable; and

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

In summary:

Claim(s)	35 U.S.C. §	Reference(s)/Basis	Claim(s) Shown Unpatentable	Claim(s) Not shown Unpatentable
1, 2, 4–7,	103	Agarwal	1, 2, 4–7, 11–14	

¹⁷ Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner's attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. § 42.8(a)(3), (b)(2).

Claim(s)	35 U.S.C. §	Reference(s)/Basis	Claim(s) Shown Unpatentable	Claim(s) Not shown Unpatentable
11–14				
1, 4, 6, 7, 11–14	103	Sinharoy	1, 4, 6, 7, 11–14	
2, 3, 16–18	103	Sinharoy, Blumrich	2, 3, 16–18	
5	103	Sinharoy, Sistla	5	
19	103	Sinharoy, Sistla, Blumrich	19	
Overall Outcome			1–7, 11–14, 16–19	

IPR2025-00207
Patent 8,984,228 B2

FOR PETITIONER:

Kevin Anderson
Patrick D. McPherson
DUANE MORRIS LLP
kpanderson@duanemorris.com
pdmcperson@duanemorris.com

Cory C. Bell
J. Preston Long
Alexander M. Boyer
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, LLP
cory.bell@finnegan.com
jp.long@finnegan.com
alexander.boyer@finnegan.com

FOR PATENT OWNER:

Tarek N. Fahmi
Jonathan Tsao
ASCENDA LAW GROUP, PC
tarek.fahmi@ascendalaw.com
jonathan.tsao@ascendalaw.com