

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Arm Ltd, MediaTek Inc.,

Petitioners

v.

Daedalus Prime LLC,

Patent Owner

Patent No. 8,984,228
Case No. IPR2025-00207

PETITIONER'S REPLY BRIEF

TABLE OF CONTENTS

I.	Introduction.....	1
II.	The Challenged Claims Are Invalid.....	1
A.	Ground 1: <i>Agarwal</i> Renders Obvious Claims 1-2, 4-7, and 11-14.....	1
1.	PO’s “Memory Regions” Argument Is Unsupported By The ‘228 Patent Claims and Specification.....	2
2.	PO Incorrectly Characterizes <i>Agarwal</i> ’s Teachings And Ignores The Petition’s Arguments.....	6
3.	The Board Correctly Already Rejected PO’s “Memory Regions” Argument On The Identical Claim Language In IPR2023-01344.....	10
4.	<i>Agarwal</i> Teaches Caching Agent Portions Associated With A Core.....	11
5.	Claim 2: PO’s TOR Arguments Are Incorrect And Conflict With The ‘228 Patent.....	13
6.	Claims 4, 5, 7, 12 and 13 Are Not Disputed.....	18
7.	Claim 6: PO Ignores The Petition Arguments And <i>Agarwal</i> ’s Disclosure.....	19
8.	Claim 11: PO Ignores The Petition’s Arguments.....	20
9.	Claim 14: PO Ignores The Petition Arguments And <i>Agarwal</i> ’s Teachings.....	21
B.	Ground 2A: <i>Sinharoy</i> Renders Obvious Claims 1, 4, 6, 7 and 11-14.....	23
1.	The Board Rejected PO’s Sparse Arguments Against <i>Sinharoy</i>	23
2.	Claim 4 Is Invalid Over <i>Sinharoy</i>	24

3.	Claim 6 Is Invalid Over <i>Sinharoy</i>	26
4.	PO Does Not Argue Claim 7 Patentability	27
5.	PO’s Arguments For Claims 11-14 Fail For The Same Reason As For Claim 6.	27
C.	Ground 2B: Claims 2, 3 and 16-18 Are Obvious.....	27
1.	Claim 3 Is Disclosed.	28
2.	Claims 16-18 Are Invalid.....	29
D.	Ground 2C: PO Mischaracterizes Both Petitioner’s Arguments And The ‘228 Patent.	29

LIST OF EXHIBITS

<u>Exhibit</u>	<u>Description</u>
Ex-1001	U.S. Patent No. 8,984,228 to Liu et al. (“the ’228 patent”).
Ex-1002	Patent Center Prosecution History of U.S. Patent No. 8,984,228.
Ex-1003	Declaration of Dr. Marwan Hassoun.
Ex-1004	Curriculum Vitae of Dr. Marwan Hassoun.
Ex-1005	U.S. Patent No. 7,805,575 to <i>Agarwal</i> et al. (“ <i>Agarwal</i> ”).
Ex-1006	B. <i>Sinharoy</i> et al., “IBM POWER7 Multicore Server Processor,” IBM Journal of Research and Development, vol. 5, no. 3, pp. 1:1-1:29 (2011) (“ <i>Sinharoy</i> ”).
Ex-1007	U.S. Patent Appl. Pub. No. 2011/0219188 to Blumrich et al. (“ <i>Blumrich</i> ”).
Ex-1008	U.S. Patent Appl. Publ. No. 2008/0126750 to Sistla (“ <i>Sistla</i> ”).
Ex-1009	<i>Multicore Processors and Systems</i> (S.W. Keckler et al. eds., 2009) (“ <i>Keckler</i> ”).
Ex-1010	U.S. Patent Appl. Pub. No. 2006/0041715 to Chrysos et al. (“ <i>Chrysos</i> ”).
Ex-1011	G. Blake et al., “A Survey of Multicore Processors,” IEEE Signal Processing Magazine, vol. 26, no. 6, pp. 26-37 (Nov. 2009) (“ <i>Blake</i> ”).
Ex-1012	V. Getov et al., “Codesign for Systems and Applications: Charting the Path to Exascale Computing,” Computer, vol. 44, no. 11, pp. 19-21 (Nov. 2011).
Ex-1013	T. Zhou, “Deriving Voltage Tolerance Specification for Processor Circuit Design,” 2011 IEEE 20th Conference on Electrical Performance of Electronic Packaging and Systems, pp. 17-20 (Oct. 2011).

<u>Exhibit</u>	<u>Description</u>
Ex-1014	Library of Congress Record: IBM Journal of Research and Development.
Ex-1015	MARC Record: IBM Journal of Research and Development.
Ex-1016	IBM Journal of Research and Development, vol. 5, no. 3 (2011).
Ex-1017	M.R. Marty, <i>Cache Coherence Techniques for Multicore Processors</i> , Ph.D. Dissertation, University of Wisconsin – Madison (2008) (“ <i>Marty</i> ”).
Ex-1018	Docket Control Order
Ex-1019	Infringement Contentions Cover
Ex-1020	IPR2023-01344 Paper 10 05/28/2024 Patent Owner’s Response
Ex-1021	IPR2023-01344 Paper 20 10/22/2024 Patent Owner’s Sur-reply
Ex-1022	Median Time To Trial
Ex-1023	Declaration of Kelley Hayes Greenfield

I. Introduction

Arm Ltd. (Petitioner) submits its Reply to the Patent Owner’s Response (“Response”). Patent Owner (“PO”) ignores the ‘228 patent’s claims and specification, the Petition’s arguments, and the prior art teachings. For example, the ‘228 patent’s Figure 6 is central to the Challenged Claims and figured prominently in the Petition. Yet, PO’s Response did not discuss or cite to Figure 6. Instead, PO manufactures a series of exclusionary positions to avoid the prior art – positions that have no basis in, and conflict with, the ‘228 patent.

As noted in the Petition (at 1), the Challenged Claim 1 elements are substantively within claim 1 of the child ‘895 patent. After the Petition (but before the Response), the Board held the child ‘895 patent’s claim 1 invalid over *Sinharoy*. IPR2023-01344, Final Written Decision, Paper 31 (January 24, 2025) (“FWD”). PO’s Response repeats arguments rejected in the FWD (as detailed below) but PO does not discuss, or distinguish, the FWD reasoning. The Board should reject PO’s arguments for the reasons in IPR2023-01344 in addition to the reasons cited in the Petition and herein.

II. The Challenged Claims Are Invalid.

A. Ground 1: *Agarwal* Renders Obvious Claims 1-2, 4-7, and 11-14

PO’s erroneous position is that the distributed caching agent portions exclude, and cannot be met by, *Agarwal*’s “distributed” cache coherency

management because *Agarwal*'s distributed directory controller / storage pairs manage particular memory regions of the overall shared cache. Response at 8-20. PO varies its language, but this is PO's fundamental position.

PO's position is incorrect for multiple reasons including: (1) it is unsupported by, and conflicts with, the '228 patent's claims and the specification; (2) it mischaracterizes *Agarwal*'s teachings and ignores the Petition's arguments; and (3) it was rejected by the Board in IPR2023-01344 for substantively identical claim language to the Challenged Claims.

1. PO's "Memory Regions" Argument Is Unsupported By The '228 Patent Claims and Specification.

PO failed to cite to the '228 patent claims or specification for PO's exclusionary meaning of distributed portions of the caching agent. By itself, PO's failure is fatal because the claims should not be interpreted (explicitly or implicitly) to exclude distributed cache coherency that manages the overall shared cache by managing individual "slices" or "partitions" or "banks" (in the '228 patent's words) which are memory regions.

PO's position conflicts with the claim language. The claims recite a "shared cache" among cores for which "cache coherency operations" are performed by the distributed caching agent portions. Thus, the claims manage the overall shared cache, not individual pieces of that cache. Claim 4 requires that the managed

shared cache portions be associated with each core. *Agarwal's* distributed cache coherency components manage in a cooperative fashion to achieve the coherency of the overall shared cache.

Figure 6 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.

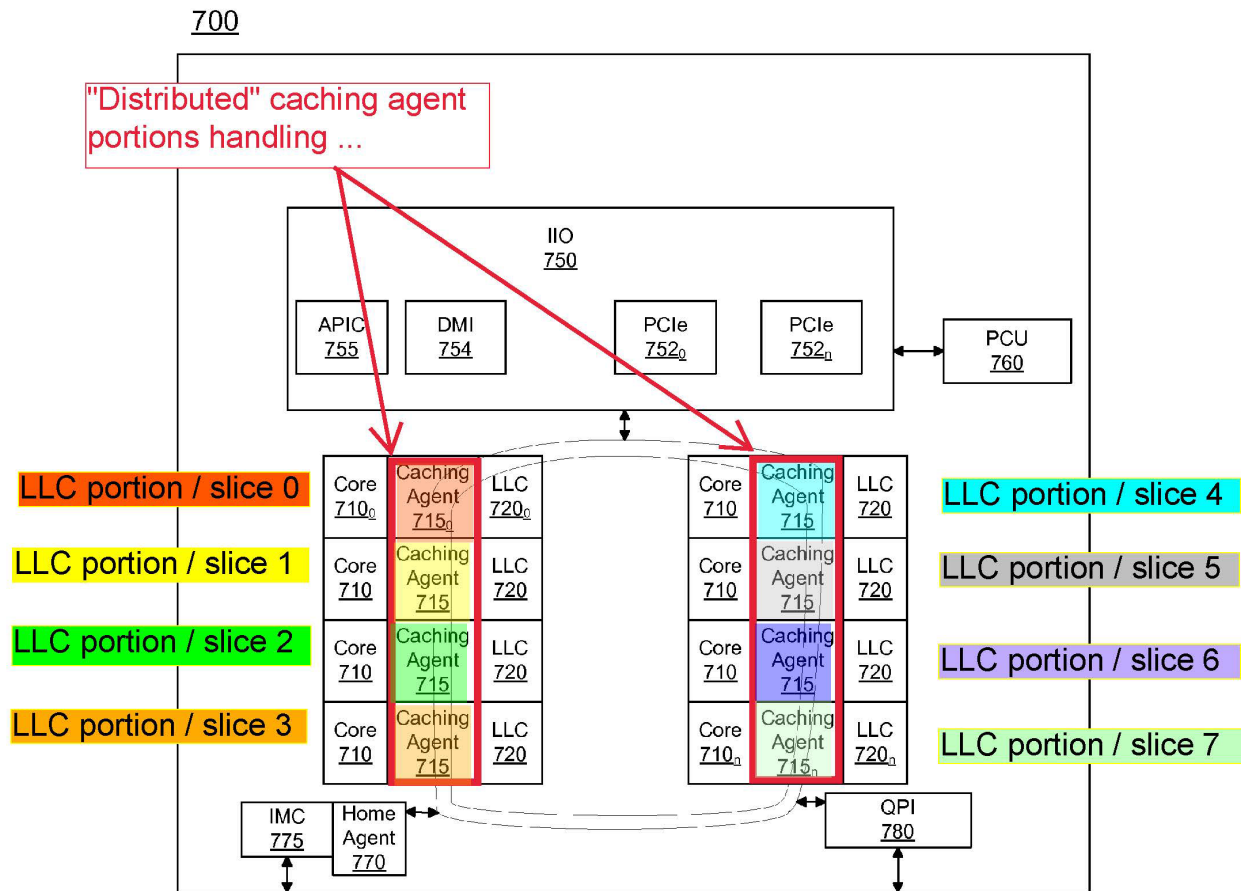


FIG. 6

(Ex-1001, Fig. 6.¹). The specification confirms that each core has a caching agent portion associated with a memory region (as in *Agarwal* and as PO seeks to exclude):

[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.

(Ex-1001, 6:22-29.). The ‘228 patent also teaches elsewhere that the caching agent portions are associated with different cache memory regions (slices / partitions) – like in *Agarwal*. *E.g.*, Ex-1001, 3:1 (“each of different portions of the caching agent can be associated with a corresponding core and LLC bank or slice”); 2:55 (“this LLC may be a bank or other portion of the LLC associated with the caching agent.”).

That caching agent 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. For example, in Figure 6, when core 710₀ (orange) accesses memory in

¹ Unless otherwise noted, all annotations and emphases herein are added.

the portion of shared LLC 720₆ (purple) associated with a different caching agent portion 715₆, the '228 patent does not restrict the cache coherency operations that occur in such access.

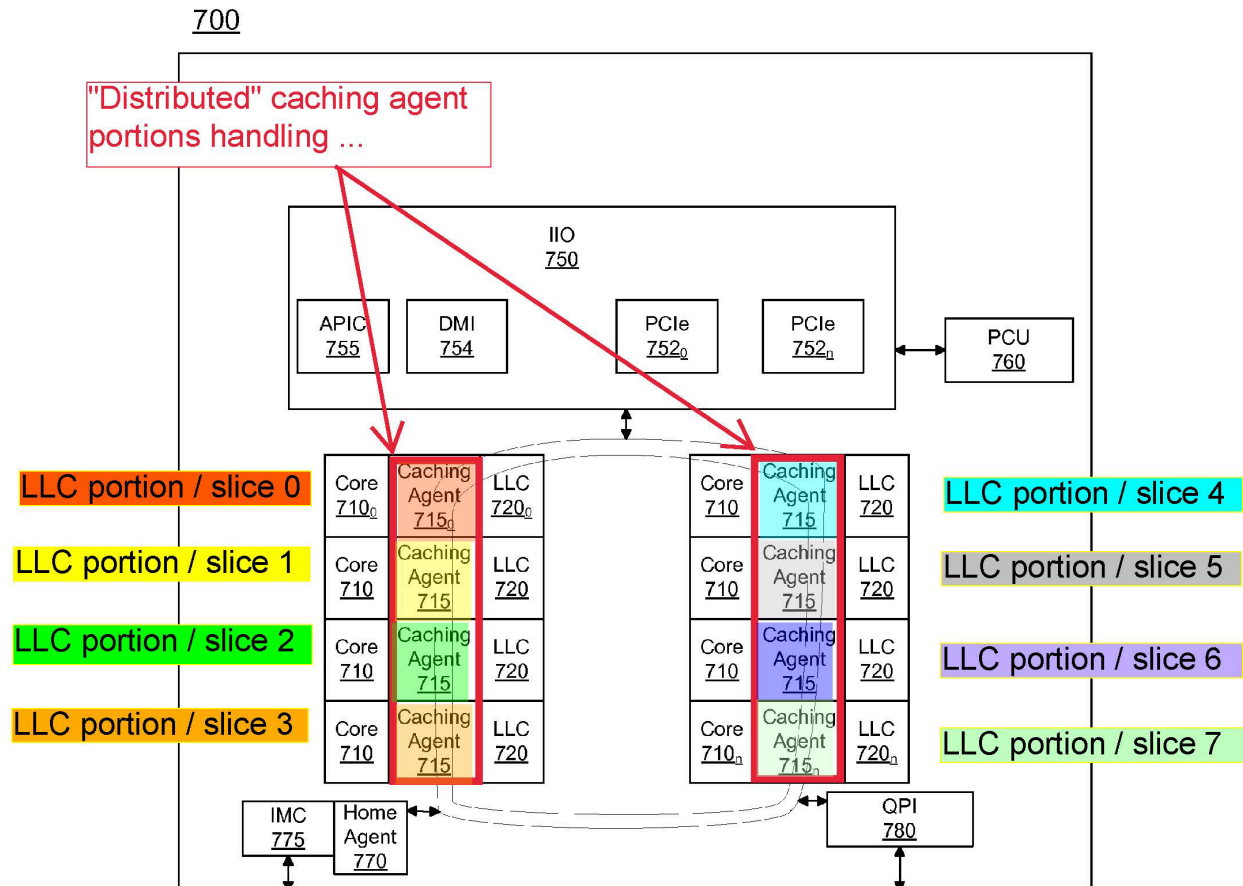


FIG. 6

PO'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. PO's Response does not discuss – or even cite – Figure 6 anywhere and yet Figure 6 is a critical disclosure of the claimed invention.

PO also appears to assert that each distributed caching agent portion must

participate in all caching operations occurring at each core – that no portion can perform any action independently even if the overall cache access and coherency occurs due to distributed operations. Response, 16-17. The ‘228 patent does not support any such restriction on the claims.

PO repeatedly asserts the claims require a “centralized control.” *E.g.*, Response at 12, 18. The ‘228 claims and specification do not recite any “centralized control.” For example, Figure 6 does not show any “centralized controll[er].” PO does not cite to the ‘228 patent for any “centralized control.” Instead, the ‘228 patent recites “distributed” cache management (the opposite of “centralized control”). Ex-1001, 2:65-3:3; 6:23-25 (“distributed configuration ... distributed portions ... handle cache coherency operations...”).

2. PO Incorrectly Characterizes *Agarwal*’s Teachings And Ignores The Petition’s Arguments.

Agarwal teaches distributed cache coherency operations corresponding to the claims. Petition, 6-9, 12-23. First, as PO acknowledges *Agarwal*’s distributed directory controller / storage pairs (“distributed portion [of the caching agent]”) manage coherency operations for portions of the overall shared cache – such that the collective distributed action of the controller / storage pairs manages coherency operations for the overall shared cache. The claims require managing the overall shared cache. *Agarwal* matches Figure 6 in that regard. Ex-1001, Fig. 6, 6:20-39;

Petition, 7, 16-23.

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. Ex-1001, 2:62-65.

Second, *Agarwal* teaches additional cooperation between caching agent portions. For example, *Agarwal* teaches mechanisms for the cores and I/O units to determine which distributed cache management portion manages a particular memory access so that requests and coherency operations can be directed correctly. These mechanisms include circuitry implementing a hash function and an address mapping table (AMT) or TLB to identify the distributed portion managing a particular address (and other types of comparable circuitry). *Agarwal*, 17:42-18:24; 27:40-59; 30:5-26; Petition 16-17, 19-23.

Third, *Agarwal*'s distributed directory coherence protocols such as the "PCC" (Response at 16, 18; Petition, 7-10, 16-23) mirror protocols used by the '228 patent. Ex-1001, 4:1-4 ("Conventional cache coherency protocol operations and compliance with ordering rules can be performed when accessing this data to maintain coherency.") 3:61-63 ("data can be stored coherently in accordance with a given cache coherency protocol."). *Agarwal* teaches such "cache protocol operations" (*Agarwal*, 1:46-2:42.). Thus, the '228 patent is consistent with, uses,

and does not exclude the type of protocols used in *Agarwal* to effectuate cache coherency operations across distributed caching agent portions (directory controllers) and the shared cache portions.

Fourth, PO disparages *Agarwal*'s "message passing," "distributed coordination," and "communication" between the distributed directory controllers. Response at 16-20. However, such communications: (a) show distributed cache management; (b) are not precluded anywhere in the '228 patent; and (c) are used by the '228 patent. *E.g.*, Ex-1001, Fig. 3, 3:33-4:50 (showing communications and messages passed in the form of allocation transactions, and "requests"). For example, the '228 patent notes that the CPU "agents" will receive "requests" which facilitate the cache management – just as in *Agarwal*. Ex-1001, 4:65-5:65.

The '228 patent uses for cache management similar "transactions" and "messages" as used in *Agarwal* that PO disparages. For example, in the '228 patent, "a read transaction in which a source agent requests a read of data" (corresponding to *Agarwal*'s cache directory controllers in tiles that issue requests) and then a "completion message provides the requested data." Ex-1001, 5:10-13. This matches *Agarwal*'s tile requesting cache data from a portion of shared cache managed by another tile and receiving a completion message with the data managed by the other tile. Response at 11 (citing *Agarwal* at 16:26-45; 19:7-20:25; 23:51-53).

Moreover, as detailed for claims 2 and 3 below, the '228 patent teaches a table of requests ("TOR") in each core within each distributed caching agent portion to store requests managed by the caching agent portion associated with that core. *E.g.*, Ex-1001, Fig. 5 (showing TOR 444_n of caching agent 440_n within each core); 5:22-48. Each core (along with its distributed caching agent that includes the TOR) receives "inbound" requests and posts "outbound" requests to other cores. *Id.* These inbound / outbound requests mirror *Agarwal*'s messages and distributed coordination.

If the '228 claims require any level of "centralized control" (not found in the '228 claims or specification), *Agarwal*'s collective, distributed actions that manage coherency for the entire shared cache provide such centralized control through their common, communicative circuitry (including the directory controllers / storage) implementing the ability to identify the distributed cache portions, communicate with the circuitry managing that identified cache portion, use known cache coherency protocols and communicate the memory requests, and results, to the requesting core. Contrary to PO's position, *Agarwal*'s cooperative directory controllers are not managing individual caches. Rather, *Agarwal*'s collective distributed protocols and components manage the entire shared cache in a distributed fashion – using the architecture and functionality described (minimally) in the '228 patent.

3. The Board Correctly Already Rejected PO’s “Memory Regions” Argument On The Identical Claim Language In IPR2023-01344.

PO raised – and lost – its argument that distributed caching agent portions that manage cache region portions do not meet the claim language. *E.g.*, IPR2023-01344, Hearing Transcript, Paper 30 (December 17, 2024) at 41:3-8 (“There is no one caching agent in *Sinharoy*... Each L3 region and each L2 has its own associated controller. And those controllers are managing the cached data within their own respective regions.”); 43:11-44:4; 47:22-26; 50:5-8; Ex-1020 at 1 (“each [prior art] L3 cache region has its own controller”), 16-17; Ex-1020 at 7 (“each cache region has its own associated controller...”); 11 (“separate cache controllers for each L3 region.”).

The IPR2023-01344 FWD considered and rejected PO’s arguments.

The claim term is “single caching agent,” not a single cache controller, and the “single caching agent” “includes a plurality of distributed portions associated with a corresponding one of the plurality of cores” like each of *Sinharoy*’s local L3 regions that share capacity, data flow logic, and control flow logic with one another and their associated L2 cache for a particular core. Ex. 1003, 19; *see* Ex. 1001, 3:6–10 (a single caching agent can be distributed so each different portion of the caching agent is associated with a corresponding core and last level cache).

FWD at 27-28.

PO does not discuss the IPR2023-01344 FWD – much less identify errors therein. This reasoning applies equally to the substantively identical ‘228 claims, in addition to *Agarwal*’s other disclosures. For example, “each of [*Agarwal*’s] local L3 regions [] share capacity, data flow logic, and control flow logic with one another” as detailed above and in the Petition. IPR2023-01344, FWD at 27-28.

4. *Agarwal* Teaches Caching Agent Portions Associated With A Core.

PO’s “associated with” argument is erroneous. Factually, *Agarwal* teaches and claims that its caching agent portions are “associated with” cores. “At least some of the directory controllers are associated with a processor core...” (*Agarwal*, 3:42-44, claim 9; Petition, 23-24). PO fails to mention *Agarwal*’s language. Response, 12-14. *Agarwal*’s explicit language meets this element.

Legally, PO manufactures a restriction of “associated with” unsupported by the claims and the ‘228 patent specification. According to PO, the “associated with” requires that the “distributed portion ... performs cache coherency operations for that core.” Response at 13. PO’s position is apparently that one distributed portion (and only that distributed portion) perform cache coherency operations for only its associated core. PO asserts that, when another core’s caching agent assists in the coherency, such assistance necessarily does not correspond with the “associated with” element. Response at 13-14.

Notably, PO does not cite language in the ‘228 patent supporting such a restrictive re-defining of “associated with.” PO’s position conflicts with the ‘228 disclosure in which no single caching agent portion does all the coherency operations for a particular core but rather acts “for the cores.” Ex-1001, 6:22-29. As shown in Figure 6 (and claim 4), different shared cache portions are associated with different cores. If core 1 accesses a memory location in shared LLC that is associated with core 4, core 4 would manage the coherency operations for its own LLC portion. Yet PO’s restrictive definition requires core 1 to manage core 4’s LLC coherency operations (which is not only nonsensical, but such cross-core management is not described in the ‘228 patent).

PO re-argues that *Agarwal*’s directory controllers are associated with memory regions (addressed above). Figure 6 depicts the claimed “single caching agent” with distributed portions and the cores and LLCs (memory regions) all associated with each other – shown as 710₀/715₀/720₀ to 710_n/715_n/720_n.

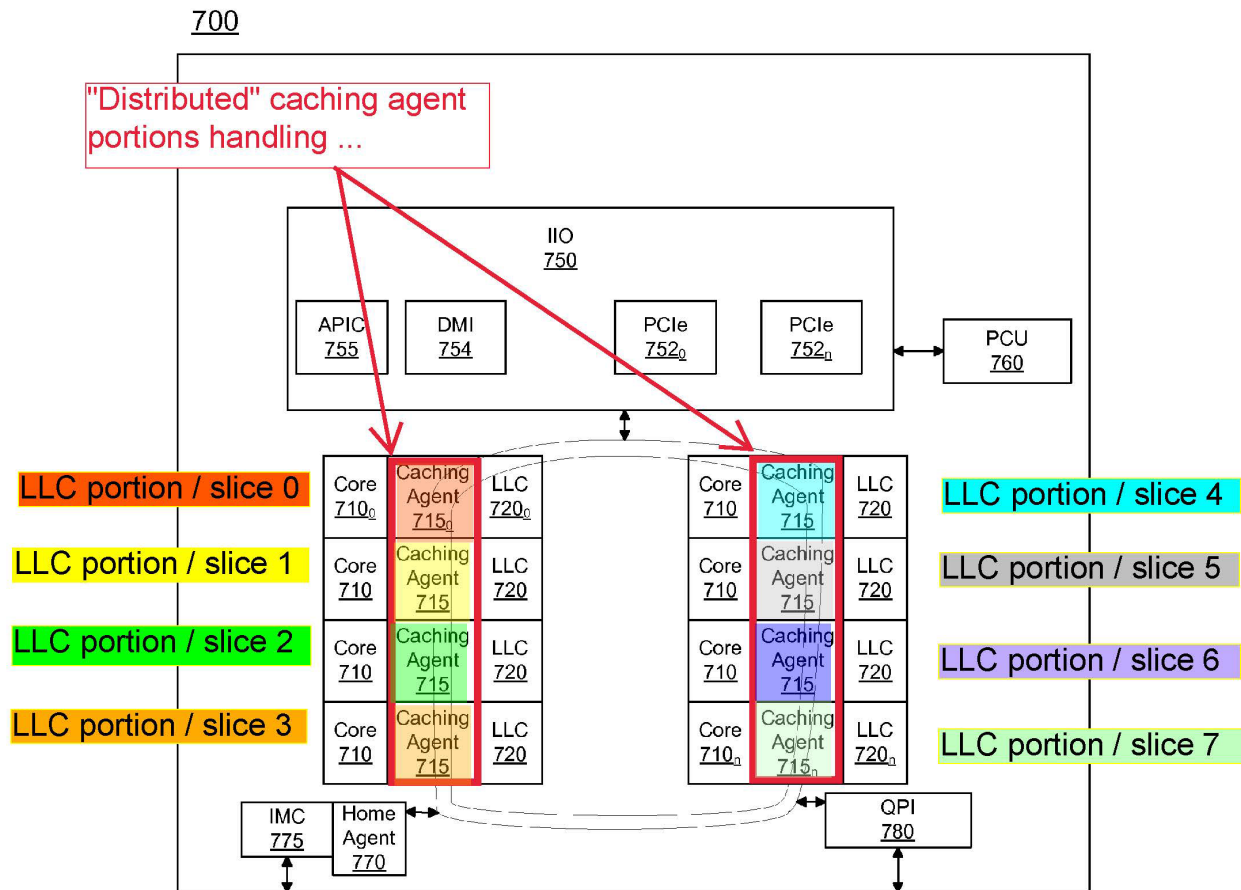


FIG. 6

(Ex-1001, Fig. 6., 6:22-29 (quoted above); *see also* Ex-1001, claim 4; 2:54-55; 3:1.

PO's position is incorrect, requires the Board to ignore *Agarwal*'s express statements, and requires the Board to ignore the '228 patent and *Agarwal*'s overlap thereof.

5. Claim 2: PO's TOR Arguments Are Incorrect And Conflict With The '228 Patent.

PO improperly demands that the TOR be a "single, shared resource" and that there can only be a "singular" TOR which is a "unified, centralized structure" or a

“shared structure... not a local buffer.” Response at 22-27 (*Agarwal*), 42-48 (*Sinharoy/Blumrich*). PO’s demand to impose a single TOR structure, to the exclusion of all other structures, is not supported by, and conflicts with, the ‘228 patent.

PO alleges “Figure 2 shows the TOR is a single, unified table within the caching agent.” Response at 22. Figure 2 does not show the “distributed” caching agent recited by the claims. Rather, 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.” Ex-1001, 2:67-3:3. Nothing in the ‘228 patent says that the TOR caching agent component is ripped out of the caching agent’s distributed portions such that there is only a single TOR.

Figure 2 shows the many components of the non-distributed caching agent
200.

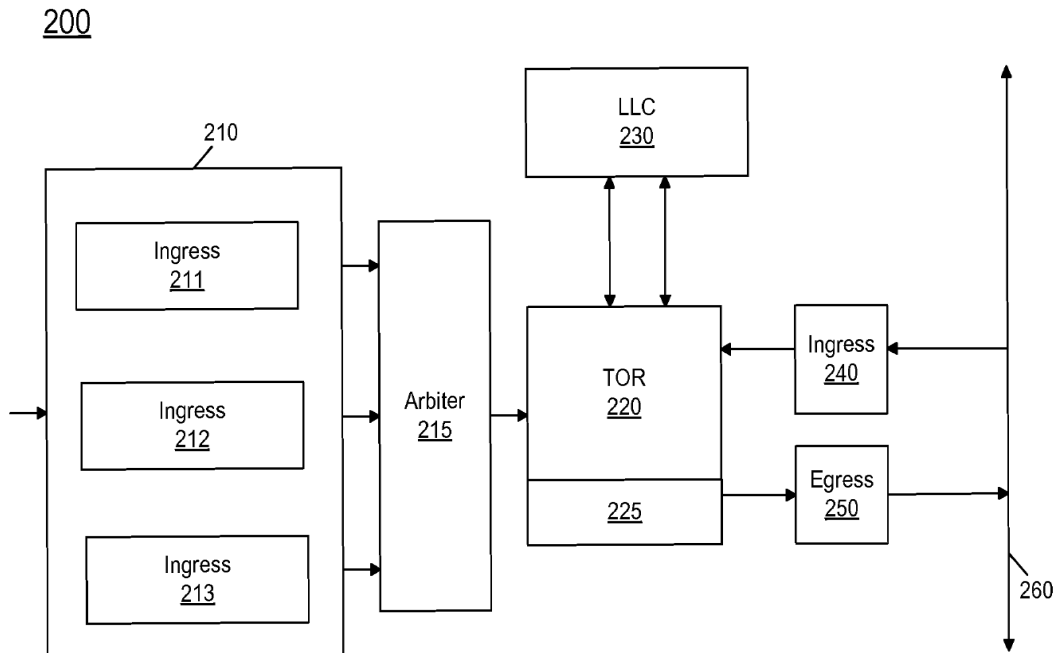


FIG. 2

Ex-1001, Fig. 2; 2:36-3:2. The '228 patent does not state (and PO does not cite) that only some of Figure 2's caching agent structures are distributed across different cores nor does the '228 patent disclose how such a single-TOR configuration would function.

Figure 6 shows the distributed caching agent version in which "each of the different portions of the [distributed] caching agent [] associated with a corresponding core."

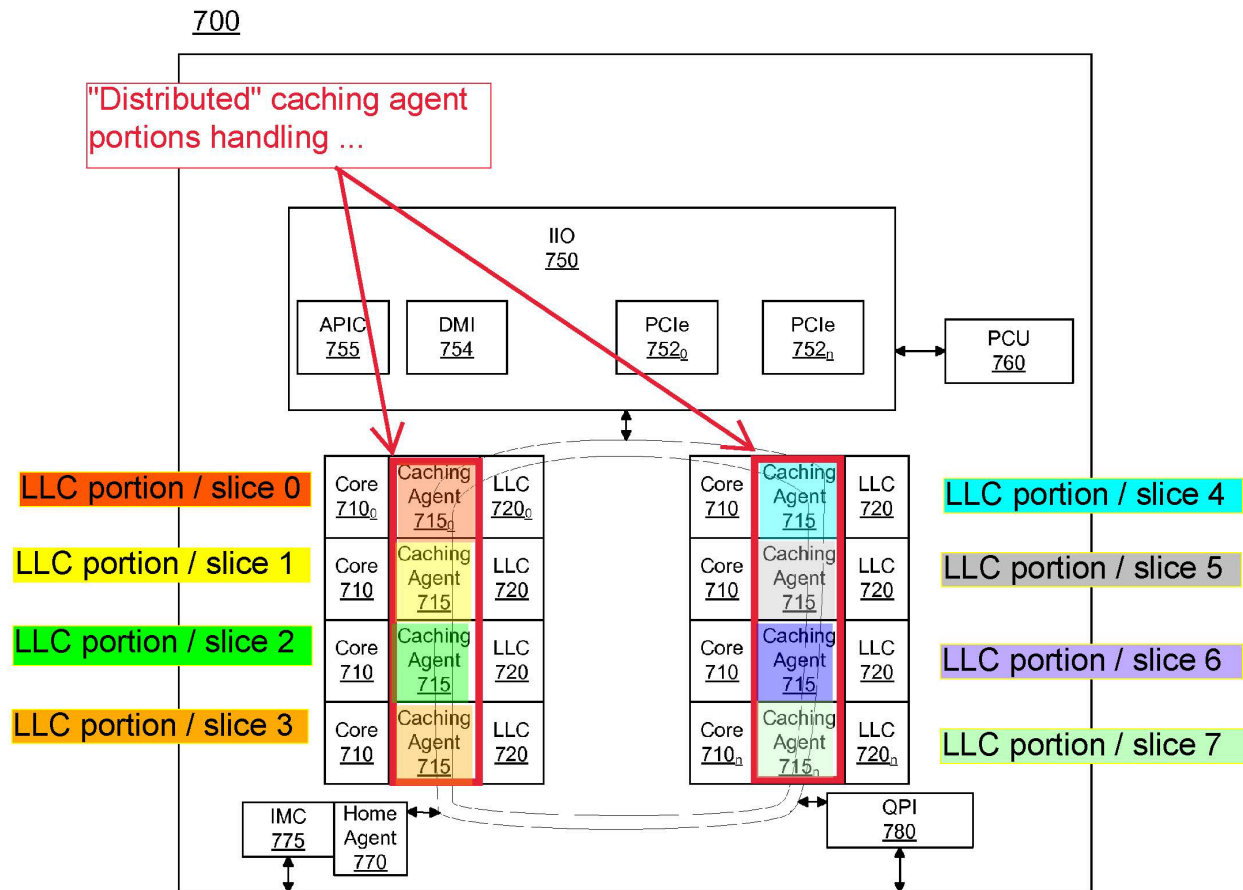


FIG. 6

Ex-1001, Fig. 6, 1:46-57, 6:20-59. Caching agents 715₀-715_n correspond to the distributed portion of Figure 2, including each structure within the caching agent.

Moreover, the '228 patent expressly teaches that the TOR is distributed across, and included in, each core in Figure 5 which shows a separate TOR in each caching agent in each core:

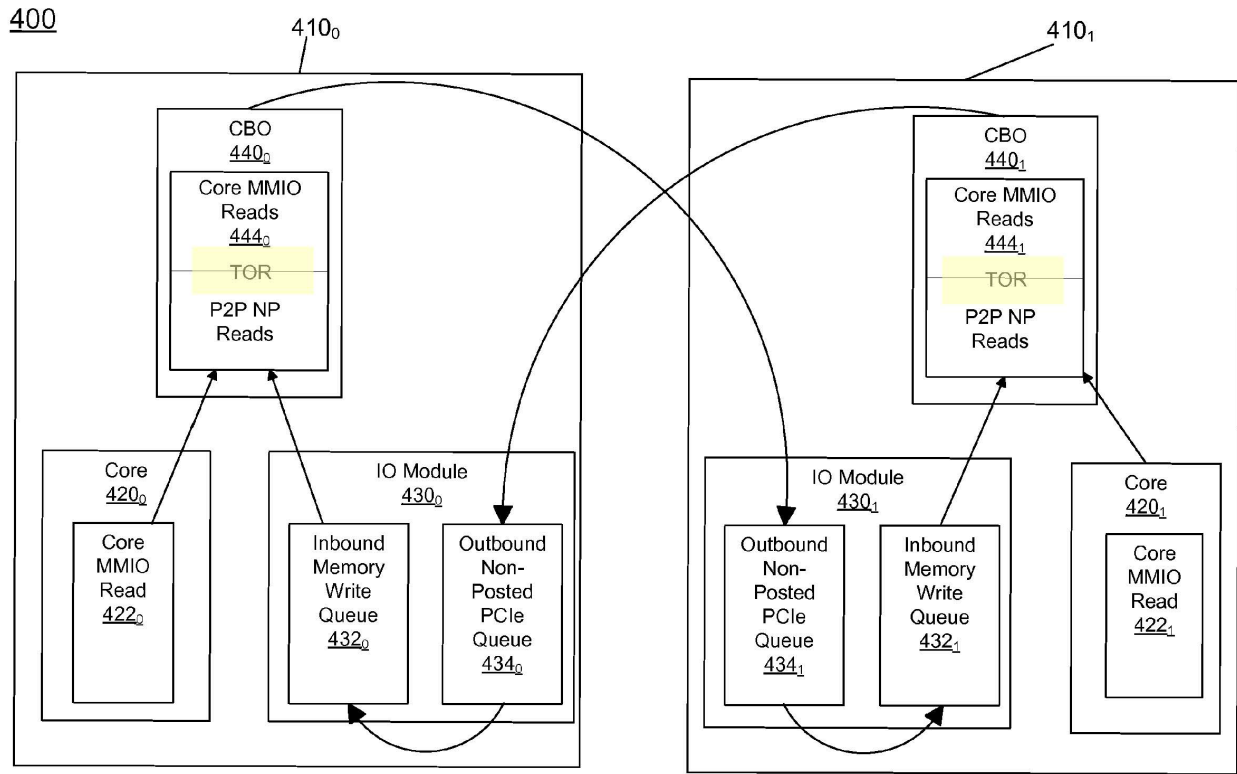


FIG. 5

Ex-1001, Fig. 5. There is a separate TOR in each core. “[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.” Ex-1001, 5:31-33. The subscript “₀” associated with the TOR 444₀ (and the other caching components) confirms that (as depicted in the Figure), there is a separate TOR for each core (such in Figure 5 as 444₀ and 444₁).

Moreover, 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. Ex-1001, 2:42-44.

Those “queues” are replicated for each core. Ex-1001, 5:31-49 (referencing repeatedly the queues specific to each core).

Furthermore, that Figure 2 shows that the TOR communicates directly with the LLC (as shown above) and Figure 6 (and claim 4) shows the LLC portions / slices distributed among the different cores, is further evidence that the TOR is also distributed among the cores (so that it can communicate with each individual LLC slice).

Notably, PO does not cite, and the ‘228 patent does not contain, any disclosure of a single “unified, centralized” TOR when the remaining components of the caching agent are distributed across the cores.

Agarwal discloses a distributed TOR akin to the ‘228 patent’s disclosure (for distributed caching agents). PO acknowledges that *Agarwal* discloses a “small buffer” that is a “per-tile” (e.g., per-core) “structure that tracks outstanding transactions for that tile’s directory controller.” Response at 22-23. This “per-tile” (per-core) structure is the distributed TOR described throughout the ‘228 patent.

Tellingly, PO does not discuss Figure 5 or its associated text. The Board should reject PO’s attempt to impose an unwarranted limitation on the TOR.

6. Claims 4, 5, 7, 12 and 13 Are Not Disputed.

PO does not argue separate patentability for claims 4, 5, 7, 12 and 13.

**7. Claim 6: PO Ignores The Petition Arguments And
Agarwal's Disclosure.**

PO's arguments on claim 6 ignore the Petition. Response at 27-29. PO incorrectly argues that "*Agarwal*'s 'cache coherence case' concerns core-initiated DMA transactions, not transactions initiated by external IO device.'" Response, 28. PO is plainly incorrect. *Agarwal*'s "cache coherence case" includes transactions from both cores and IO devices. Petition, 28-30, 16-23.

First, *Agarwal* teaches the claimed "allocation transaction" from an IO device. "An I/O module is also able to ... store data directly into the cache of a given tile...." (*Agarwal*, 28:23-32.) This satisfies the claims irrespective of PO's arguments.

Second, *Agarwal* expressly states that the IO devices are using the "cache coherence case."

The situation that occurs when the I/O module 912 writes directly into a tile is case 2: the "**cache coherence case.**" *Agarwal*, 28:31-32. PO's arguments conflict with *Agarwal*'s express language.

Third, PO misleadingly quotes *Agarwal*'s statement that "direct memory access (DMA) transactions *initiated from a tile...*" Response at 28 (emphasis in original) (quoting *Agarwal*, 28:10-22). The language quoted by PO relates to a different option in *Agarwal* than the option cited in the Petition.

Agarwal's very next paragraph describes a different option relied upon by

the Petition in which “a[n] I/O module is also able to process a DMA transaction or otherwise store data directly into the cache of a given tile.... The situation that occurs when the I/O module 912 writes directly into a tile is case 2: the ‘cache coherence case.’” *Agarwal*, 28:23-32.

Because PO manufactured its own misleading strawman argument, the Petition’s arguments are un rebutted (and meet claim 6).

8. Claim 11: PO Ignores The Petition’s Arguments

For claim 11, PO ignores the Petition and *Agarwal* to make two incorrect arguments. First, PO reiterates its claim 6 argument that *Agarwal* does not disclose an “‘allocation transaction’ ... *from an IO device*....” Response at 29-30 (emphasis in original). As detailed for claim 6, PO is incorrect.

Second, PO cites claim language of “forwarding the allocation transaction ... to a [shared] caching agent...” to regurgitate PO’s incorrect attack on *Agarwal*’s disclosure of a distributed single caching agent in claim 1. Response at 30-31. For claim 11, PO repeats its assertion that *Agarwal* solely relies upon “whichever home tile is associated with the target memory address.” Response, 31. As detailed above for claim 1, PO’s attack on *Agarwal* is inconsistent with the ‘228 patent, ignores *Agarwal*’s disclosure, and in conflict with the Board’s decision in IPR2023-01344.

PO crafts an argument alleging a “proxying IIO traffic” requirement.

Response, 31. First, claim 1 does not have any proxy requirement. Second, PO does not cite to any passage in the '228 patent requiring any such proxy requirement. Response at 31-32. PO cites to paragraph 100 of its expert declaration which cites (but does not discuss or explain) "EX. 1001, 3:45-4:31." Response at 31, citing Ex-2006, ¶100.

The passage at 3:45-4:31 explains Figure 3. Neither that passage, nor Figure 3, teaches any such proxy requirement. Relevant to claim 11, the specification mirrors the 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.

The Petition details this forwarding at pp. 32-33 (citing claims 6-7) and in claims 6-7 at pp. 28-31. PO does not address (much less rebut) any Petition argument for "forwarding." As with its claim 6 arguments, PO ignores that *Agarwal* teaches that Figure 9C is the "ProxyIO coherence protocol" which shows the "cache coherence case" in which a shared, distributed cache coherence agent across the IIO module and cores manages the cache collectively for the shared cache. *Agarwal*, 29:3-30:26, 28:23-29:2 (describing the "cache coherence case."). PO's arguments should be rejected for the same reasons as for claims 6, 7 and 1.

**9. Claim 14: PO Ignores The Petition Arguments And
Agarwal's Teachings.**

PO ignores the Petition and *Agarwal* to incorrectly argue that *Agarwal* does

not disclose the negative limitation of “directly storing data ... without communicating with the system memory.” Claim 14 overlaps with claims 6, 7 and 12. Thus, the Petition relied upon those claims’ analysis. Petition, 37.

PO argues that “[o]bviousness requires more than silence.” Response, 33. But “silence” is not what *Agarwal* teaches. Rather, as detailed for claims 7 and 12, *Agarwal* teaches two distinct cases: the “memory coherence case” (Fig. 9B) and the “cache coherence case” (Fig. 9C) In *Agarwal*’s “cache coherence case” in Fig. 9C, 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. *Agarwal*, 29:5-30:11.

Thus, this is not “silence” but rather a juxtaposition of different scenarios with an express reference to a memory access in the “memory coherence case” and no reference to a memory access in the “cache coherence case.” This juxtaposition of two distinct scenarios teaches, or at a minimum, renders obvious that the cache coherence case does not write into the system memory as is done in the memory coherence case because no transaction to system memory is shown.

PO would impose a restriction that the prior art (such as *Agarwal*) must “affirmatively” recite each action that it does not take. No such requirement exists for patent law.

Moreover, *Agarwal* identifies situations for the Figure 9C “cache coherency

case” in which cache stores occur without any system memory access. *Agarwal* notes that, in some cases when the store (write) is to only a portion of the cache line, *Agarwal*’s system handles the store without fetching (initiating a memory transaction) to achieve the “benefit of ... reduced bandwidth demand on the memory interface.”

Also, in some cases if a write transaction is occurring in a region in which the previously stored data is no longer needed, the home tile does not have to fetch background data for incomplete line writes that occur, for example at the beginning or end of a multi-line write. ... A benefit of not needing to fetch the background data is reduced bandwidth demand on the memory interface. In cases in which the previously stored data is needed, the tile can fetch the background data from main memory in the case of a partial line write.

Agarwal, 30:12-26. Thus, this passage expressly teaches that in some situations *Agarwal* writes to the cache without communicating with system memory.

B. Ground 2A: *Sinharoy* Renders Obvious Claims 1, 4, 6, 7 and 11-14.

1. The Board Rejected PO’s Sparse Arguments Against *Sinharoy*.

Claim 1 is substantively identical to the child ‘895 patent’s claim 1 that the Board held invalid over *Sinharoy* in IPR2023-01344. (Petition, 1-2.) PO does not mention, much less distinguish, the IPR2023-01344 FWD holding that *Sinharoy* discloses the language in claim 1. Thus, under both the reasoning in IPR2023-

01344 and basic collateral estoppel principles, claim 1 is invalid over *Sinharoy*.

PO presents a half-hearted, brief argument on claim 1 using the reasoning rejected in IPR2023-01344. Response at 35-36. PO's argument is that "*Sinharoy* [allegedly] employs multiple independent controllers" and thus does not disclose a "single agent managing coherency for both CPU and I/O domains." Response at 36. The Board addressed this exact argument. IPR2023-01344, FWD at 24-28. The Board rejected PO's assertion that *Sinharoy*'s L3 cache controllers were "independent" for different L3 cache regions. IPR2023-01344, FWD at 26-27.

PO regurgitates its "snoop" argument (Response at 36), but the Board properly rejected that argument also. IPR2023-01344, 28 ("The caches act in concert as a group, i.e., a 'single caching agent,' to solve that miss" by one of the cores).

Notably, PO does not address the Petition's detailed explanation that *Sinharoy* discloses the single caching agent. Petition, 2-4 (explaining '228 patent "single caching agent"), 46-57 (detailing *Sinharoy*'s invalidating disclosure). The Petition's analysis stands un rebutted and provides additional reasons for the invalidity of claim 1 beyond the analysis in IPR2023-01344.

2. Claim 4 Is Invalid Over *Sinharoy*

Claim 4 matches the child '895 patent's claim 5 which was held invalid in IPR2023-01344. IPR2023-01344, FWD at 29. Thus, PO is estopped from re-

challenging that *Sinharoy* discloses this claim.

PO's arguments are inconsistent with the claim language and the '228 patent specification. Response at 37-40. PO asserts that "*Sinharoy* fails to disclose [the claimed] association" but PO re-defines "associated" inconsistent with the specification. *Supra* at §II.A.4. According to PO, in *Sinharoy* "each L3 region is accessible by any of the cores." Response at 38.

The claim term recites "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" and does not require that each LLC portion be "accessible" by only a single core – and does not preclude that each core can access each region / portion. The point of a "shared cache memory" is that the memory is "shared" among the cores – not limited in access to any particular core.

PO's arguments conflict with the '228 patent's embodiments describing the LLC as shared and accessible by all cores. *E.g.*, Ex-1001, 3:65-4:1 ("With the data now present in the LLC, it can be accessed a number of times for read and write operations by any of the cores of the processor as well as the initiating IO device"); 6:32-36; 2:15-17; 3:13-18 ("any processor core within the same socket can have a very fast path to access the data" "push[ed]" into the LLC); *see also* Ex-1001, 6:20-39; 3:45-4:1.

Furthermore, Figure 6 shows the "distributed" caching agent recited in the

Challenged Claims and depicts separate portions of LLC associated with each of those caching agent portions. Yet PO ignores Figure 6. Ex-1001, Fig. 6; 6:20-39.

3. Claim 6 Is Invalid Over *Sinharoy*

Claim 6 recites the same “allocation transaction” element recited in the child ‘895 patent’s claim 1 which was held invalid in IPR2023-01344. IPR2023-01344, FWD at 24-29. Thus, PO is estopped from re-challenging that *Sinharoy* discloses this claim.

PO incorrectly argues that *Sinharoy*’s “cache injection ... directed to the correct physical L3 slice based on an address-homing function” to imply that there is no cache coherency operation for the cache injection. Response at 40 (citing *Sinharoy* (*Sinharoy* at 17) and Dr. Hassoun’s deposition at EX-2007, 8:4-81:16). Neither *Sinharoy* nor Dr. Hassoun’s testimony state what PO alleges.

Rather, as in the Petition (at 57-58), *Sinharoy*’s “cache injection” is the same type of “coherent operations” as for other cache operations:

... new **coherent** operations are supported. One such operation is cache injection. An I/O device performing a direct memory access (DMA) write operation may target the operation to the cache, instead of to memory.

Sinharoy, 17. Because the “cache injection” is a “coherent operation” on the cache as other accesses, there is no basis to conclude (as PO argues) that *Sinharoy*’s caching agent does not receive, and process, the cache injections in the manner as

other coherent operations. *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. Petition, 57-58. This shows the "intermediate recipient" cited by PO. PO's remaining arguments are a half-hearted re-hash of its arguments for claim 1 that *Sinharoy* does not disclose the shared caching agent.

4. PO Does Not Argue Claim 7 Patentability

PO does not argue claim 7.

5. PO's Arguments For Claims 11-14 Fail For The Same Reason As For Claim 6.

For claims 11-14, PO cites its claim 6 arguments – which fail for the reasons discussed in claim 6.

C. Ground 2B: Claims 2, 3 and 16-18 Are Obvious

Claim 2 matches the child '895 patent's claim 3 which was held invalid in IPR2023-01344. IPR2023-01344, FWD at 29. Thus, PO is estopped from re-challenging that *Sinharoy* discloses this claim.

As detailed for *Agarwal*, PO's assertion that the TOR in claim 2 must be a "unified" or "centralized" structure across all cores is incorrect and conflicts with the '228 patent's embodiments. *Supra* at §II.A,5. Thus, PO's arguments fail because the premise (of a single TOR for all cores) is incorrect. Response at 42-48.

PO argues that the “specialized RC and write machines manage different transaction types (read/claim vs. write).” Response at 43. Factually, the Petition asserts that the collection of the RC and write machines teach the TOR. Thus, the mere fact that *Sinharoy* might have separate entries for read and writes does not mean that *Sinharoy* does not disclose the TOR. Such dedicated entries are claimed in claim 3. As PO acknowledges, the TOR “dedicated entry” in claim 3 is dedicated to “posted” transactions which PO alleges are “one-way, non-completion I/O writes (‘posted’).” Response at 46. Thus, *Sinharoy*’s teaching of different entries for reads and writes mirrors the ‘228 patent and matches claim 3. Legally, claim 2 requires only a “table of requests.” It does not require that the table include both read and write requests.

PO does not dispute that *Sinharoy*’s structure holds caching operations for core and IIO requests. Response at 44-46. Thus, PO’s argument that Blumrich does not disclose a buffer for “mixed core/IIO requests” misses the point – that functionality is already in *Sinharoy* and is not disputed by PO. Response at 44.

1. Claim 3 Is Disclosed.

Claim 3 matches the child ‘895 patent’s claim 4 which was held invalid in IPR2023-01344. IPR2023-01344, FWD at 29. Thus, PO is estopped from re-challenging that *Sinharoy* discloses this claim.

As detailed for claim 2, PO argued that *Sinharoy* has separate structures for

read and write transactions. Response at 43. Given this acknowledgement, PO cannot challenge that *Sinharoy* (and *Sinharoy-Blumrich*) disclose Claim 3. PO's also reargues its incorrect "unified" TOR position.

As stated above, the RC and write machines within *Sinharoy*'s caching agent include temporary storages ("TOR") to store incoming read / write requests. When write machines perform cache injection operations, they write into L3 cache. (*Sinharoy*, 17, 19.) This is a "posted transaction" because *Sinharoy* does not disclose any "completion or confirmation message regarding the" injection. (Ex-1001, 5:3-7) (Ex-1003 ¶477.)

Because write machines within *Sinharoy*'s caching agent have temporary storages, it would be obvious to include dedicated queues or resources (i.e., dedicated entry) to store such a write transaction (i.e., posted transaction). (Ex-1003 ¶478.) Thus, if *Sinharoy* alone does not render claim 3 obvious, the *Sinharoy-Blumrich* combination renders claim 3 obvious. (Ex-1003 ¶479.)

2. Claims 16-18 Are Invalid.

PO relies on its erroneous arguments for claims 2 and 3 for claims 16-18.

D. Ground 2C: PO Mischaracterizes Both Petitioner's Arguments And The '228 Patent.

PO's claim 5 argument is that "Sistla's 'ring' couples clusters, not distributed portions of a caching agent." Response at 50-51. First, PO's argument

mischaracterizes the claims to imply (or assert) that the ring interconnect can connect only the caching agent – and nothing else. Claims 5 and 19 do not recite any such restriction.

Moreover, the ‘228 patent’s disclosure of “ring interconnect” confirms that it connects all portions of the cores (just like Sistla):

As seen, a ring interconnect 730 thus couples the cores together, and provides interconnection between the cores, caching agent 715 and IIO module 750.

Ex-1001, 6:36-39; Fig. 6. Thus, PO’s argument improperly limits the claims to exclude connecting all the components.

Second, PO mischaracterizes the Petition which stated that *Sinharoy*’s caching agents were connected by the ring interconnect. “*Sinharoy*’s 4-MB L3 regions including their controllers ...would be coupled via a ring interconnect.” Petition, 77.

PO crafts an argument that “Petitioner’s theory of obviousness depends on conflating Sistla’s SA with the caching agent recited in Claims 1 and 19.” (Response at 51). The Petition noted that Sistla’s SA is a relevant feature and that the relevant features in such cores are compatible with a ring interconnect. At best, PO quarrels that Sistla’s SA “is not a ‘caching agent’ as recited in the challenged claims.” Response at 51. But that is not Petitioner’s argument. Rather, *Sinharoy*

discloses the caching agent; Sistla provides a known mechanism (ring interconnect) for connecting components. The Petition provided additional arguments supporting the combination – arguments that PO did not dispute. Petition, 76-78.

March 18, 2026

Respectfully submitted,

Kevin P. Anderson

USPTO Reg. No. 43,471

CERTIFICATION OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, Petitioners certify that the word count of Petitioners' Reply for *Inter Partes* Review (exclusive of any table of contents, table of authorities, mandatory notices under § 42.8, certificates of service and word count, and list of exhibits or claim listing) as measured by Microsoft Word is 5581.

Dated: March 19, 2026

By: Kevin P. Anderson
Kevin P. Anderson, Lead Counsel
Reg. No. 43,471

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(a), the undersigned certifies that on March 19, 2026 a copy of the foregoing Reply for *Inter Partes* Review, the was served Pursuant to 37 CFR § 42.105 and agreement on counsel via electronic download to the following attorneys:

Tarek Fahmi

tarek.fahmi@ascendallaw.com

Date: March 19, 2026

By: Kevin P. Anderson
Kevin P. Anderson