

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

INTEL CORPORATION,
Petitioner,

v.

ADVANCED CLUSTER SYSTEMS, INC.,
Patent Owner.

IPR2025-00915
U.S. Patent No. 11,563,621

SUPPLEMENTAL DECLARATION OF CHANDRAJIT L. BAJAJ, PH.D.

TABLE OF CONTENTS

I.	INTRODUCTION.....	3
II.	PRIOR ART ANALYSIS	4
	A. Menon and Trefethen teach a <i>first cluster node module configured to: receive first commands from a user interface without the first commands first passing through the first kernel</i>	4
	1. Menon and Trefethen teach the claimed “user interface flow path”	7
	B. Menon and Trefethen teach that <i>the first cluster node module is logically positioned between the user interface and the first kernel</i>	9
	1. The claim language “ <i>logically positioned between</i> ” does not exclude intermediate components	10
	C. Menon, Trefethen, and MPIref teach the claimed transmission of tasks without transmitting the tasks through a master node.....	11
	D. RS6000 teaches that “the first processor and the second processor reside on the same die” as recited in claims 20 and 25	14
	1. The claim term “reside” does not mean “integrated in” or “fabricated on” as Patent Owner argues	16
	2. The “POWER3” disclosure confirms obviousness under Patent Owner’s overly-narrow “single-chip” interpretation	17
	E. Combining Menon and Trefethen would have been obvious to a POSITA	19
	1. Menon and Trefethen teach how to implement MultiMATLAB.....	19
	2. Menon does not teach away from the combination	20
	F. A POSITA would have known how to locate the cited technical materials	23
III.	DECLARATION	24

I. INTRODUCTION

1. I, Chandrajit L. Bajaj, have been retained by counsel for Intel Corporation (“Petitioner”) as a technical expert in connection with the proceeding identified above. I submit this declaration in support of Petitioner’s Reply in pending *Inter Partes* Review (“IPR”) IPR2025-00915 of claims 1-30 (the “Challenged Claims”) of U.S. Patent No. 11,563,621 (“the ’621 patent”).

2. My qualifications are summarized in my prior declaration in this case (Ex.1003, ¶¶7-17) and in my CV (Ex.1004).

3. I am being compensated for my work in this matter at an hourly rate. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this matter. My compensation is not contingent on the outcome of this matter or the specifics of my testimony. I have no personal or financial stake or interest in the outcome of this proceeding.

4. I have been asked to provide opinions regarding specific issues raised in Patent Owner’s Response (Paper 21, “POR”) and in Petitioner’s Reply being filed concurrently with this declaration.

5. In the preparation of this declaration, I have considered all materials listed in my prior declaration (Ex.1003, ¶4), the materials listed below, and any other materials cited in this declaration:

- (1) Decision Granting Institution of Inter Partes Review (Paper 15);
- (2) Patent Owner’s Response (Paper 21);

- (3) U.S. Patent No. 6,234,829 (“Thomsen”), Ex.1164;
- (4) U.S. Patent No. 6,178,526 (“Nguyen”), Ex.1165;
- (5) U.S. Patent No. 6,003,100 (“Lee”), Ex.1166;
- (6) U.S. Patent No. 5,856,937 (“Chu”), Ex.1167;

6. In forming my opinions, I also have considered: the relevant legal standards, including the standard for obviousness, which Intel’s counsel has explained to me and which are described in my prior declaration (*see* Ex.1003, ¶¶18-23); and my own knowledge and experience based upon my work in the field of cluster computing, including computer cluster systems and cluster computing techniques, and the background additional materials cited in this declaration.

7. This declaration applies the same level of ordinary skill in the art described in my prior declaration. *See* Ex.1003, ¶¶24-27.

8. Throughout my analysis below, I have used *italics* to indicate claim language, and **bolded** text in quoted material to indicate emphasis. Unless otherwise noted, all emphasis in any quoted material has been added.

II. PRIOR ART ANALYSIS

- A. Menon and Trefethen teach a *first cluster node module configured to: receive first commands from a user interface without the first commands first passing through the first kernel*

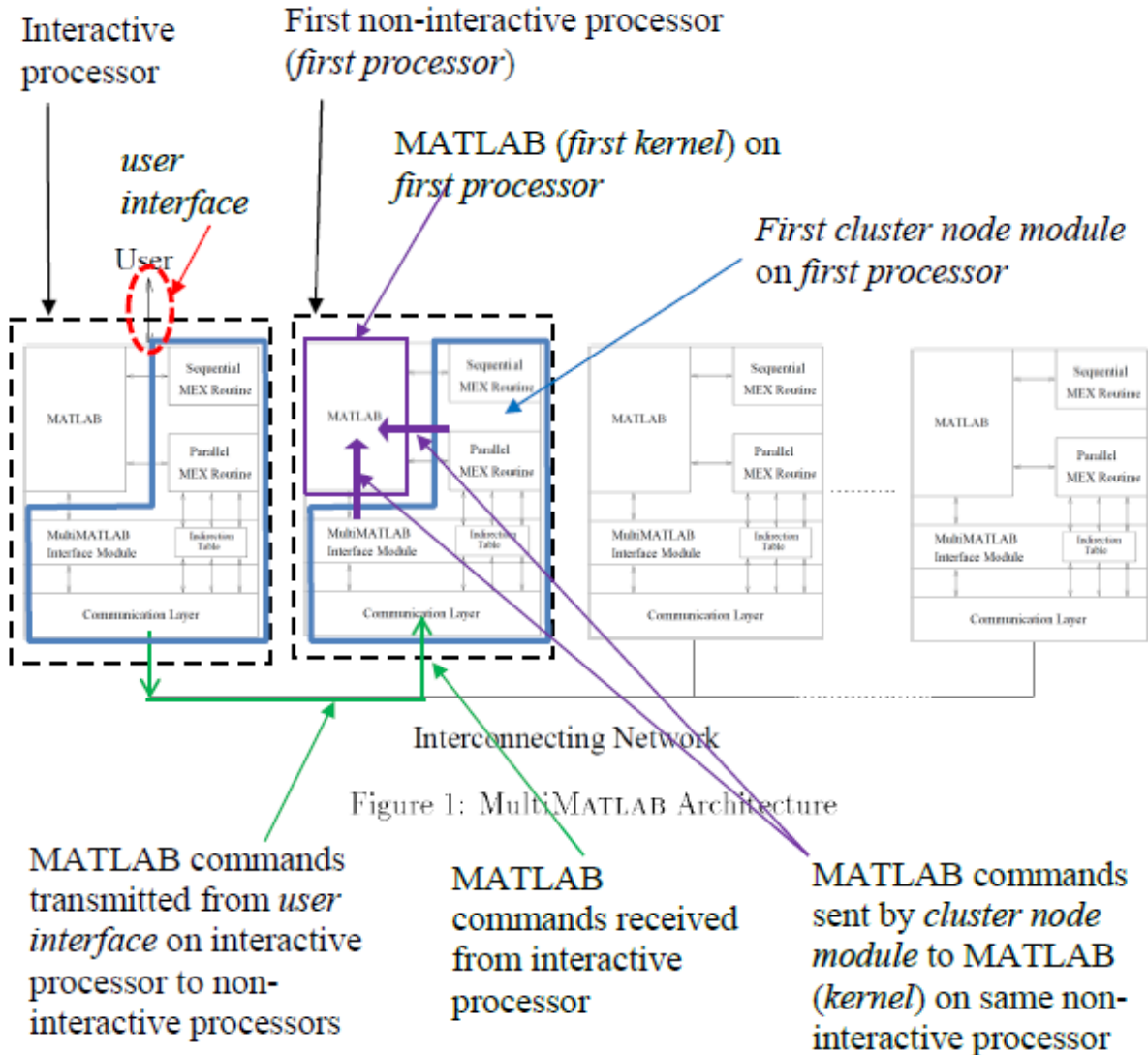
9. Claims 1 and 13 recite *a first cluster node module*. As described in my prior declaration, Menon discloses that the MATLAB process address space located in memory includes a set of components to provide interconnection with other

processes: “MEX Routines,” “MultiMATLAB interface module,” “indirection table” and “communication layer,” which collectively teach the *first cluster node module*. Ex.1005, 3, 5; Ex.1003, ¶¶200-205.

10. Claim 1 further recites that the first cluster node module is configured to: receive first commands from a user interface without the first commands first passing through the first kernel. In Menon, a “user interacts directly with one MATLAB process, called the interactive process” to have user instructions executed in parallel by the other processor nodes. Ex.1005, 3; Ex.1003, ¶209. The interactive processor has “an intuitive graphical interface” to allow users to “specify geometries, differential equations, and boundary conditions.” Ex.1005, 6; Ex.1003, ¶209.

11. Further, Menon and Trefethen teach that the user enters commands (*first commands*) that are received by the “first non-interactive processor” directly from the “interactive processor” without first passing through the “MATLAB process” on the “first non-interactive processor” (*the first kernel*). See Ex.1006, 3; Ex.1005, 3-4, 6; Ex.1003, ¶210. Menon and Trefethen teach that “the routine Eval”—which is a “MEX routine” and thus part of the *first cluster node module*—“accepts” MATLAB commands and “sends the command[s]” to the MATLAB process on each non-interactive processor of MultiMATLAB. Ex.1005, 3-4, 6; Ex.1006, 3. For example, “the user now types Eval(‘sqrt(2)’),” which causes “the

MATLAB command `sqrt(2)` [to be] executed on all six processors.” Ex.1006, 3;
Ex.1003, ¶210. The following figure shows this interaction:



Ex.1005, FIGs. 1 and 2 (combined and annotated); Ex.1003, ¶212

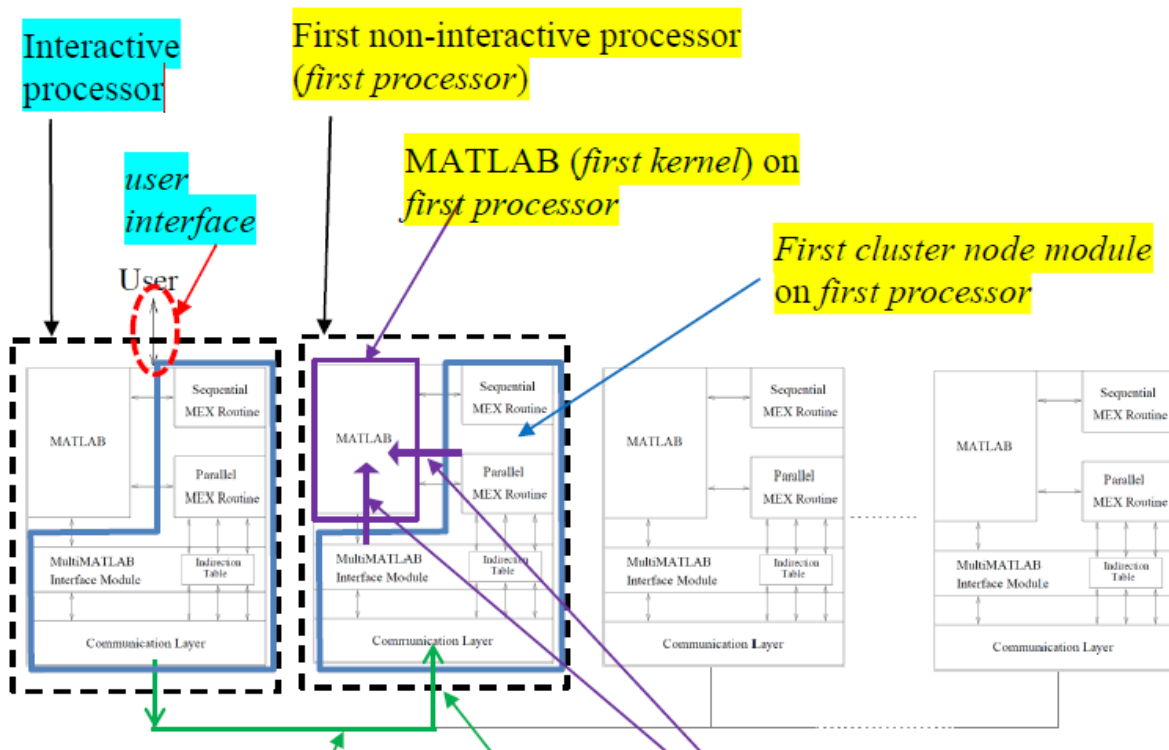
12. In my opinion, Menon and Trefethen teach that (i) the user enters a command via the user interface on the interactive processor, (ii) the cluster node module on the interactive processor communicates the command to the cluster node

modules on the first processor, second processor and third processor, (iii) the cluster node module of the first processor receives the command via the interconnecting network (as depicted by the green arrow), and (iv) the cluster node module of the first processor communicates the command to the first kernel (as depicted by the purple arrow). Ex.1006, 3; Ex.1005, 3-4, 6; Ex.1003, ¶215-220. So Menon and Trefethen teach that the command entered by the user is received by the first cluster node module without the command first passing through the first kernel. A POSITA would understand the command to be “from” the user interface because the user interface is the origin of the command; the claim does not require a direct physical connection from the user interface to the first cluster node module and does not forbid the command from passing through components other than the first kernel.

1. Menon and Trefethen teach the claimed “user interface flow path”

13. I understand that Patent Owner argues that Menon and Trefethen do not “describe the user interface flow path” and do not “provide the details as to what components of the Interactive Process the user communicates.” POR, 12-16. The claim limitation is directed to the first cluster node module on the first non-interactive processor receiving commands from the user interface without those commands first passing through the first kernel; it does not require identifying every component inside the interactive process that handles the user's keystrokes or graphical-input events. As explained above, in the Menon-Trefethen combination,

the command originates with the user interface of the interactive process and then is sent to the first non-interactive process's cluster node module before being passed to the first kernel. The following figure shows this mapping, with components from the interactive process highlighted blue, and components from the first non-interactive process highlighted yellow:



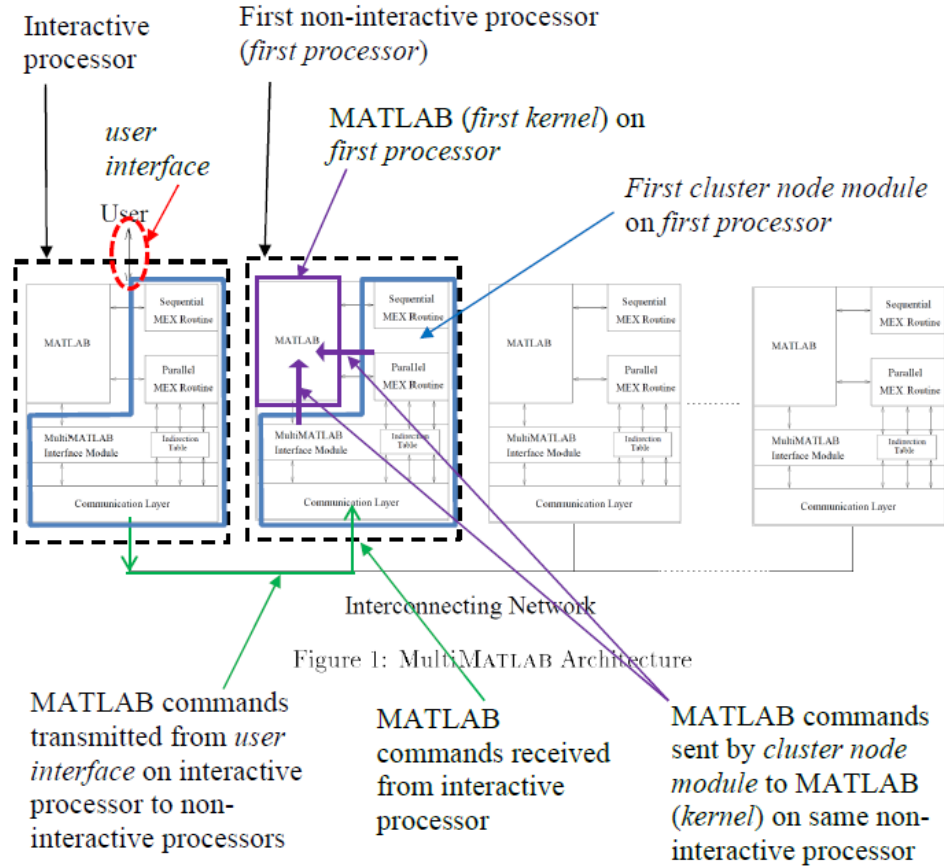
Ex.1005, FIGs. 1 and 2 (combined and annotated); Ex.1003, ¶212

14. Accordingly, additional implementation details of the user-interface path within the interactive process are not necessary to understand the mapping of Menon and Trefethen to claim 1, because the first kernel and first cluster node module are mapped to components of the first non-interactive process, not the interactive process. See Ex.1006, 3; Ex.1005, 3-4, 6; Ex.1003, ¶215-220. What

matters for the claim is that the command reaches the first cluster node module before it reaches the first kernel, and Menon and Trefethen teach that ordering.

B. Menon and Trefethen teach that *the first cluster node module is logically positioned between the user interface and the first kernel*

15. Limitations [2.0], [14.0], [24.4.3], and [27.4.3] recite that “*the first cluster node module is logically positioned between the/a user interface and the first kernel.*” My prior declaration shows that Menon and Trefethen teach these portions of the claims. Ex.1003, ¶¶287-288. In my prior declaration, referring to the combination figure of Figures 1 and 2 of Menon reproduced below, I stated that “the MEX routines of *the first cluster node module* in a MATLAB process on the first non-interactive processor are operationally positioned between (*is logically positioned between*) the user interface on the interactive processor and the MATLAB on the first non-interactive processor (*the first kernel*).” Ex.1003, ¶288. The depicted operational flow is (i) the user enters commands via the user interface (as depicted in red in the figure below), (ii) those commands are handled by the cluster node module of the interactive processor and transmitted via the network (as depicted in green), (iii) the commands are received by the cluster node module of the first processor (as depicted in green sending the commands to the blue cluster node module), and the (iv) cluster node module of the first processor sends the commands to the first kernel (as depicted in purple). So the cluster node module of the first processor is positioned between the user interface and the first kernel.



Ex.1005, FIGs. 1 and 2 (combined and annotated); Ex.1003, ¶288

1. The claim language “logically positioned between” does not exclude intermediate components

16. I understand that Patent Owner argues that the “cluster node module” in the first non-interactive processor, i.e., the *first cluster node module*, “is logically positioned between the first kernel and the first Interactive Processor—not between the first kernel and the user interface.” POR, 18. Patent Owner argues that because “[t]he Interactive Processor is an intermediary between the user interface and the first non-interactive processor and its components,” the “logically positioned between” limitation is not met by Menon and Trefethen. POR, 18. I disagree. The

presence of such an “intermediary” component does not preclude the components of the first non-interactive processor mapped to the first cluster node module from being “*logically positioned between*” those mapped to the *first kernel* and the *user interface*. A POSITA would understand “logically positioned between” to describe the functional order of components in the command path, not to require immediate adjacency or to exclude any other component from also being on that path. In my prior declaration, I interpret this arrangement in Menon and Trefethen, including the supposed “intermediary” component, as the first cluster node module being “*logically positioned between*” the first kernel and the user interface, consistent with how a POSITA would have understood such an arrangement of components. See Ex.1003, ¶¶287-288. The interactive process is upstream of the first cluster node module; it does not sit between the first cluster node module and the first kernel on the first processor.

C. Menon, Trefethen, and MPIref teach the claimed transmission of tasks without transmitting the tasks through a master node

17. I understand that Patent Owner argues that MultiMATLAB is a “master-controlled” architecture in which worker processes never transmit tasks to each other and all task routing flows through the interactive process. POR, 3-4, 26-27, 34-36. I disagree. Patent Owner conflates how a user initiates an overall computation with how the cluster node modules communicate tasks and data once the MultiMATLAB processes are running.

18. As explained in my prior declaration, each MATLAB process in Menon includes MEX routines, the MultiMATLAB interface module, an indirection table, and a communication layer, which I map to a cluster node module. Ex.1005, 3-5; Ex.1003, ¶¶200-205, 261-273. Menon explains that each process has the ability to communicate with other processes through a communication layer over the interconnection network and that all parallel functionality is provided through MEX routines. Ex.1005, 3-5. A POSITA would understand those teachings to describe communication capability at each process, not only at the interactive process.

19. Trefethen provides concrete examples of non-interactive processes executing Send and Recv commands to exchange data with other non-interactive processes, including a cyclic message-passing example in which data is passed from process to process. Ex.1006, 5-7; Ex.1003, ¶¶261-273. In those examples, the interactive process may issue an Eval command that causes the non-interactive processes to execute the relevant Send/Recv code, but the subsequent process-to-process messages are not transmitted through the interactive process as a master node. Rather, the local cluster node module on the sending process invokes the communication layer/MPI to communicate over the network with the receiving process's cluster node module.

20. The claim language requires that the cluster node modules “can transmit tasks” without transmitting those tasks through a master node. It does not require that a worker process autonomously decide, without any user-initiated command, to begin a computation. A POSITA would understand that a task can be initiated by a user command and still be transmitted peer-to-peer between cluster node modules without going through a master node. This is what Menon, Trefethen, and MPIref teach in combination.

21. I also understand Patent Owner criticizes my opinions regarding the claimed data structure for messages originating from other cluster node modules. POR, 7, 34-36. My prior declaration explains that the MEX Recv routines allocate memory for received data, that Menon and Trefethen describe received MATLAB data structures such as matrices, cell arrays, and structures, and that MPIref discloses receive buffers/queues associated with MPI receive operations. Ex.1005, 9-10; Ex.1006, 10; Ex.1017, 15-21; Ex.1003, ¶¶274-286. A POSITA would understand those buffers and allocated MATLAB data structures to be data structures in which messages/data originating from other cluster node modules are stored.

D. RS6000 teaches that “the first processor and the second processor reside on the same die” as recited in claims 20 and 25

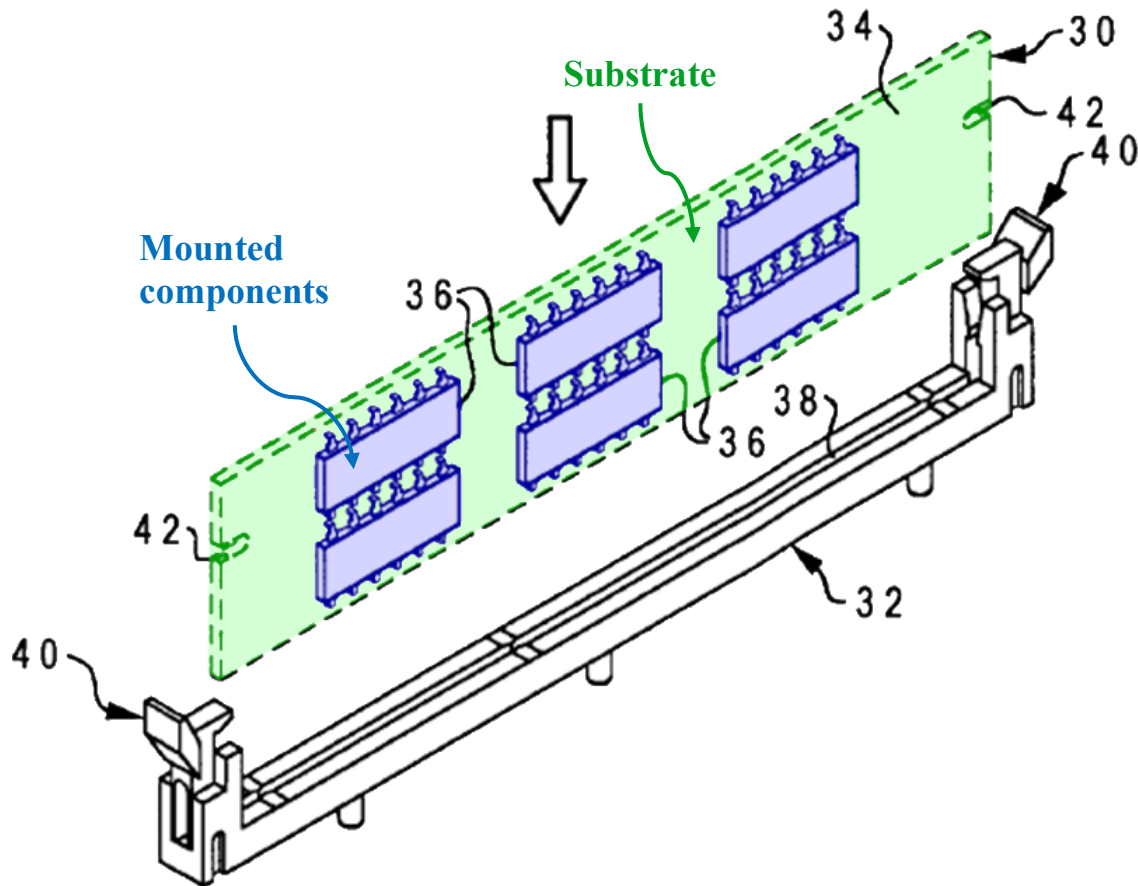
22. Claims 20 and 25 of the '621 patent recites that “*the first processor and the second processor reside on the same die.*” Ex.1001, claims 20, 25. In the Menon-Trefethen-RS6000 combination, RS6000 teaches this limitation by disclosing “**four 375 MHz, 630FP processors**” (including *the first processor and the second processor*) “**mounted**” (*reside*) “**on one card**” (*on the same die*, e.g. a single semiconductor substrate). Ex.1007, 10; Ex.1003, ¶385. In my opinion, this disclosure demonstrates that a POSITA would have understood the mapped processors to reside together on a single processor-node substrate/assembly, rather than on separate computers or separate master/worker systems. To the extent Patent Owner argues that “same die” requires a single semiconductor chip, I address that narrower interpretation below.

23. I understand that Patent Owner equates the “same die” language with a “single semiconductor substrate” and then argues that the processors must be fabricated together on that substrate. POR, 4, 18-22. A POSITA would understand that components can “reside on” a substrate or card by being mounted or located there, and the '621 specification uses “reside” in that locational sense, as discussed below.

24. Several examples from the prior art confirm that a card or board was commonly described as a substrate on which components reside or are mounted.

For example, Thomsen describes “a flat, rectangular **substrate or card 34**” on which components are “mounted,” as shown in the following annotated FIG. 2.

Ex.1164, 1:62-64, FIG. 2; *see also*, e.g., 3:17-18, 3:54-57.



Ex.1164 (Thomsen), Detail of FIG. 2 (annotated)

25. Other references confirm the same usage. Nguyen describes a “PC motherboard” as a “substrate” on which “components” such as a “microprocessor” are mounted. Ex.1165, 2:7-22. Lee describes a “printed circuit board” as a “substrate” on which a “central processing unit (CPU)” is mounted. Ex.1166, 5:15-18. Chu describes a processor module as having a “printed-circuit board substrate”

and explains that a CPU or microprocessor may be mounted on a processor card/module. Ex.1167, 1:51-54, 2:17-29. These references confirm that a POSITA would not have read “reside on” as synonymous with “fabricated in” or “integrated into.”

1. The claim term “reside” does not mean “integrated in” or “fabricated on” as Patent Owner argues

26. I understand that Patent Owner interprets the claims’ recitation that the processors *reside on the same die* to mean that the processors must be “**fabricated together** on a single semiconductor substrate.” POR, 4, 20 (processors *resid[ing] on the same die* means “**fabricating** multiple processors on a single silicon substrate”). This interpretation of the term “**reside**” is not consistent with the ’621 specification’s use of the term, or with what a POSITA would have understood the term to mean. The ’621 specification consistently uses the term “reside” to describe a current location of a component or data, which was consistent with the plain and ordinary meaning of the term to a POSITA. *See, e.g.*, Ex.1001, 2:34-61, 11:6-9, 11:27-29, 22:64-67.

27. For example, the ’621 specification describes that the “communications network 102 may connect to the computer systems 110, 120, 130, for example, by use of a modem or by use of **a network interface card that resides in each of the systems.**” Ex.1001, 11:6-9. A POSITA would have understood this disclosure to describe a network interface card that is currently

installed or located within each of the computer systems rather than any aspect of the card's "fabrication" as the POR argues. *See id.*; POR, 4, 20.

28. The '621 specification also describes "**kernel modules**" that "**reside in the memory of one or more computer systems on which they run.**" Ex.1001, 11:27-29; *see also, e.g.*, claim 3 ("the first kernel and the first cluster node module **reside** in the first computer-readable medium"), 22:64-67. Again, a POSITA would have understood this disclosure to describe the current storage location of particular data (e.g., "kernel modules") rather than anything related to "fabrication" as the POR argues. *See id.*; POR, 4, 20.

29. So the '621 specification consistently uses the term "**reside**" to describe the current location of components or data with a computer system, which is consistent with the term's plain and ordinary meaning. *See, e.g.*, Ex.1001, 2:34-61, 11:6-9, 11:27-29, 22:64-67. Patent Owner's interpretation of the term as requiring particular "fabrication" details is inconsistent with both the term's usage in the '621 specification and its plain and ordinary meaning to a POSITA, and is therefore improper. *See* POR, 4, 20; *see, e.g.*, Ex.1001, 2:34-61, 11:6-9, 11:27-29, 22:64-67.

2. The "POWER3" disclosure confirms obviousness under Patent Owner's overly-narrow "single-chip" interpretation

30. I understand that Patent Owner may interpret claims 20 and 25 to require a multi-core microprocessor. Even under this incorrect interpretation,

RS6000 still teaches the claimed subject matter. As discussed above, RS6000 teaches “**four 375 MHz, 630FP processors** mounted on one card.” Ex.1007, 10. Each of these four processors can be a “**375 MHz POWER3**” microprocessor. Ex.1007, 1-2, 9-12; Ex.1010, 52. Each POWER3 microprocessor “is a single chip implemented with 0.25 micron CMOS technology” that is “comprised of **eight execution units**...allow[ing] **concurrent operation** of ... instructions.” Ex.1010, 57. The ’621 specification also describes “processing cores” as “**units capable** of single-threaded **execution**[.]” Ex.1001, 7:27-29. In my opinion, the single POWER3 chip having “eight execution units” teaches a single chip package having 8 processing cores (which are defined by ’621 specification as a “unit[] capable of ... execution”). Ex.1010, 57; Ex.1001, 7:27-29.

31. I do not rely on the label “execution unit” as a one-to-one synonym for a modern commercial “core.” Rather, POWER3 confirms that, before the ’621 patent’s priority date, processor designers integrated multiple concurrently operating execution resources on a single chip. A POSITA considering RS6000’s multi-processor node and seeking a compact, high-performance implementation would have understood single-chip integration of multiple processor execution resources to be a known and predictable design option. This further confirms that Patent Owner’s Core 2 Duo-centered reading is too narrow for the language and specification of the ’621 patent. Ex.1010, 57; Ex.1001, 7:27-29.

32. Moreover, a POSITA would understand that POWER3's multiple execution units, which allow concurrent operation of instructions, are consistent with the specification's functional discussion of processing resources and contradict Patent Owner's attempt to confine the claim language to a single two-core commercial microprocessor example. Ex.1010, 57; Ex.1001, 7:27-29. I do not rely on this point as a substitute for the SMP mapping discussed above; rather, it confirms that Dr. Smith's single-chip, Core 2 Duo-centered reading is too narrow for the language and specification of the '034 patent.

E. Combining Menon and Trefethen would have been obvious to a POSITA

1. Menon and Trefethen teach how to implement MultiMATLAB

33. My prior declaration (Ex.1003, ¶¶65-73) describes multiple reasons why a POSITA would have been motivated to combine Menon and Trefethen, including:

- Menon suggesting the combination by **expressly citing** Trefethen and explaining that Menon builds on the teachings of Trefethen. Ex.1005, 1-2; Ex.1003, ¶71.
- Menon and Trefethen both describe aspects of how the same MultiMATLAB system enables MATLAB to perform parallel mathematical calculations. Ex.1005, Abstract, 16; Ex.1006, 2; Ex.1003, ¶72.

- Menon and Trefethen together represent a combination of prior art elements according to known methods (executing MATLAB code on a multi-node system) to yield the predictable result discussed in both references: the MultiMATLAB system executing MATLAB in parallel to perform mathematical calculations. Ex.1005, Abstract; Ex.1006, 2-7; Ex.1003, ¶70.

34. A POSITA would have had an especially strong reason to read these references together because they address the same project, the same MATLAB-based parallel-computing problem, and the same MultiMATLAB command environment. Ex.1005, 1-2; Ex.1006, 2-7. Menon expressly identifies Trefethen’s MultiMATLAB work and then describes a redesigned architecture that preserves the same user-level objective—parallel MATLAB computations—while improving implementation details and extensibility. Ex.1005, 2, 14.

35. A POSITA therefore would not view the references as mutually exclusive alternatives. Rather, a POSITA would understand Trefethen as teaching specific MultiMATLAB commands and examples, including Eval, Send, Recv, and the cycle example, and Menon as teaching how the MultiMATLAB architecture was implemented and improved in a later version. Ex.1005, 2-4, 14; Ex.1006, 4-7.

2. Menon does not teach away from the combination

36. I understand that Patent Owner argues that Menon “teaches away” from Trefethen because Menon describes the “MultiMATLAB architecture” of

Trefethen as “different from the newer one described in” Menon. POR, 42-44. I disagree.

37. Menon does not “distinguish” the “MultiMATLAB” teachings of Trefethen. Menon explicitly cites Trefethen and explains that Menon builds on the teachings of Trefethen. Ex.1005, 1-2; Ex.1003, ¶71. Menon describes *enhancing* the system of Trefethen. Ex.1005, 2. For example, Menon describes a list of “goals” already achieved by MultiMATLAB and then states that the architecture has been “redesigned to achieve two **further** goals.” Ex.1005, 2. One of these two “further goals” is “extensibility.” Menon achieves this extensibility by adding “MEX routines” to the system of Trefethen. Ex.1005, 2-3. The “MEX routines” do not show an “incompatibility” between Menon and Trefethen. Instead, Menon’s addition of the “MEX routines” to MultiMATLAB shows that such a modification was within the capabilities of a POSITA because the authors of the Menon paper actually implemented the modification. Ex.1005, 2-3.

38. In my opinion a POSITA reading Menon would not have been led to disregard the teachings of Trefethen. Menon specifically references Trefethen and incorporates the teachings of Trefethen—Menon certainly does not disregard Trefethen. Ex.1005, 1-2 (citing Trefethen). Accordingly, Menon does not teach away from the combination with Trefethen.

39. Menon’s discussion of the first architecture’s “performance limitations” and lack of support for parallel MEX extensions would not discourage a POSITA from using Trefethen’s teachings. On the contrary, Menon identifies those limitations as reasons for the redesign and then explains that the newer architecture achieves further goals. Ex.1005, 2, 14. A POSITA would understand that critique as a motivation to improve MultiMATLAB while retaining relevant teachings of the earlier work, not as a warning that Trefethen’s disclosed commands and examples should be discarded.

40. I therefore disagree with Patent Owner’s assertion that Menon teaches away. Menon does not state that Trefethen’s Eval, Send, Recv, or data-passing examples are inoperable. It cites Trefethen, discusses the same MultiMATLAB system, and explains modifications that improve the architecture. Ex.1005, 1-2, 14. In my opinion, this is evidence of a reason to combine the teachings, not evidence that a POSITA would have avoided them.

41. Patent Owner’s teaching-away argument also overlooks the specific portions of Trefethen on which I rely. I rely on Trefethen for user-level MultiMATLAB commands and examples such as Eval, Send, Recv, and cyclic process-to-process communication. Menon does not criticize those commands as inoperable. Instead, Menon preserves MultiMATLAB’s user-level objective—

parallel MATLAB computations—while improving implementation details and extensibility. Ex.1005, 2, 14; Ex.1006, 3-7.

F. A POSITA would have known how to locate the cited technical materials

42. I understand that Patent Owner also argues that Menon, Trefethen, RS6000, POEref, and MPIref were not publicly accessible. POR, 39-46. From a technical perspective, these are precisely the kinds of materials a POSITA working on MATLAB-based parallel computing and IBM SP cluster implementations would have known to consult. *See* Ex.1003, ¶¶121-142.

43. Menon was an ACM/IEEE Supercomputing paper directed to “Integrating MATLAB with High-Performance Parallel Computing.” Ex.1005, 1. Supercomputing conference proceedings were a natural source for a POSITA interested in high-performance parallel computing. Menon also expressly cites Trefethen’s MultiMATLAB work and discusses the same MultiMATLAB system. Ex.1005, 1-2. A POSITA reading Menon would therefore have been led to Trefethen for command-level examples of Eval, Send, Recv, and cycle.m.

44. Similarly, RS6000 and POEref are IBM technical materials corresponding to the IBM SP/RS6000 and POE environment that Menon expressly identifies as the platform for MultiMATLAB. Ex.1005, 5-6. A POSITA implementing or evaluating MultiMATLAB on that platform would have had a technical reason to consult those IBM materials. And MPIref is the MPI standard

that Menon expressly invokes because of MPI's popularity, suitability for high-performance parallel computing, and vendor-optimized implementations. Ex.1005, 5; Ex.1017, 1. These facts reinforce my opinion that the references are not hindsight-selected, disconnected materials.

III. DECLARATION

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: April 30, 2026_____



Chandrajit L. Bajaj, Ph.D.