

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAMSUNG ELECTRONICS CO., LTD. and
SAMSUNG ELECTRONICS AMERICA, INC.,
Petitioner,

v.

RADIAN MEMORY SYSTEMS, LLC,
Patent Owner.

Case IPR2025-01350
Patent 11,740,801

EXHIBIT 2020

DECLARATION OF DR. MARK JONES

TABLE OF CONTENTS

I.	INTRODUCTION	3
II.	QUALIFICATIONS	3
III.	BASES OF OPINION	5
IV.	APPLICABLE LEGAL STANDARDS	7
A.	Level Of Ordinary Skill In The Art.....	7
B.	My Understanding Of Legal Standards	9
V.	CLAIMS AND GROUNDS AT ISSUE	11
VI.	OPINIONS	12
A.	Petitioner’s Sinclair Combinations	12
1.	Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.....	13
a)	Petitioner Does Not Say What Its Combination Of Sinclair And Yamada Actually Is.	13
b)	Petitioner Combines Mutually Inconsistent Embodiments Of Sinclair Without Explanation.....	16
2.	Petitioner Does Not Show That Its Sinclair Combinations Teach “a maintenance request issued by the host.”	23
a)	Petitioner Does Not Show Any “maintenance request issued by a host” In Sinclair.	24
b)	Petitioner Does Not Show That Yamada Teaches Any “maintenance request issued by the host” In Its Combination.	28
3.	Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller Having Logic To “identify a specific one of the respective subdivisions”	29

4.	Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller “provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value.”	33
B.	Petitioner’s Terasaki Combination.....	41
1.	Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.....	42
a)	Petitioner Does Not Say What Its Combination Of Terasaki And Yamada Actually Is.	42
2.	Petitioner Does Not Show That Its Terasaki Combination Teaches Any “maintenance request issued by the host.”	44
a)	Petitioner Does Not Show Any “maintenance request issued by [a] host” In Terasaki.	45
b)	Petitioner Does Not Show That Yamada Teaches Any “maintenance request issued by the host” In Its Combination.	52
3.	Petitioner Does Not Show That Its Terasaki Combination Teaches Logic To Cause A Memory Controller To “identify a specific one of the respective subdivisions”	54
4.	Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller “stor[ing] a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host.”	57
5.	Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller “provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value.”	61
VII.	CONCLUSION	66

I. INTRODUCTION

1. My name is Mark Jones. I have been retained as an expert witness to provide my independent opinion in regard to the matters at issue in *inter partes* review of U.S. Patent No. 11,740,801 (“the ’801 patent”) in IPR2025-001350. I have been retained by Radian Memory Systems, LLC (“Radian” or “Patent Owner”), the Patent Owner in this proceeding. Petitioner is Samsung Electronics Co., Ltd. and Samsung Electronics America, Inc. (“Samsung” or “Petitioner”).

2. I am being compensated \$750 per hour for my time spent working in connection with this case. My compensation is in no way related to the outcome of this litigation. If called as a witness, I would testify as to the statements and opinions contained in this report.

3. I am not a legal expert and offer no opinions on the law. However, I have been informed by counsel of the various legal standards that apply, and I have applied those standards in arriving at my conclusions.

II. QUALIFICATIONS

4. I graduated summa cum laude from Clemson University in 1986 with a B.S. in Computer Science and a minor in Computer Engineering, while holding a National Merit Scholarship and the R. F. Poole Scholarship. I graduated from Duke University in 1990 with a Ph.D. in Computer Science, while holding the Von Neumann Fellowship.

5. Upon graduation, I joined the Department of Energy at its Argonne National Laboratory facility. While at the Department of Energy, I received the IEEE Gordon Bell Prize.

6. In 1994, I joined the Computer Science faculty at the University of Tennessee. While at the University of Tennessee, I received a CAREER Award from the National Science Foundation.

7. In 1997, I joined the Electrical and Computer Engineering faculty at Virginia Tech. I was cited multiple times on the College of Engineering's Dean's List for teaching.

8. In 2021, I became a Professor Emeritus of Electrical and Computer Engineering at Virginia Tech.

9. During my academic career, my teaching included courses on computer architecture and computer organization. These classes frequently covered topics specific to storage systems and more specifically to flash storage systems—including, for example, the division of responsibility for storage management between software and hardware on a host system and software and hardware on an embedded controller. My research has also at times involved the integration of flash storage into various computer architectures.

10. In addition to the activities, education, and professional experience listed above, I have been involved in research projects that contribute to my expertise

relating to this report. While at Virginia Tech, for example, I was a primary or co-investigator on government and industrial research grants and contracts in excess of five million dollars.

11. A detailed record of my professional qualifications is set forth in the attached Exhibit A, which is my *curriculum vitae*, including a list of publications, awards, research grants, and professional activities. My *curriculum vitae* also lists the matters in which I have served as an expert.

III. BASES OF OPINION

12. In the course of conducting my analysis and forming my opinions, I have reviewed materials including those listed below:

- i. The Petition filed by Petitioner in IPR2025-01350,
- ii. U.S. Patent No. 11,740,801 (“’801 Patent”) (Ex. 1001);
- iii. The prosecution history of the ’801 Patent (Ex. 1002);
- iv. The Declaration of Dr. Heiner Litz in IPR2025-01350 (Ex. 1003) (the “Litz Declaration” or “Litz-Decl.”);
- v. U.S. Publication No. 2009/0271562 A1 (“Sinclair”) (Ex. 2019),¹

¹ Although the Petition specifically identifies U.S. Publication No. 2009/0271562 A1 as “Sinclair,” and although Petitioner refers to Sinclair as “Ex. 1004,” Petitioner appears to have provided a copy of a different reference to the

- vi. U.S. Patent Application Publication No. 2011/0271032 (“Yamada”) (Ex. 1005);
- vii. U.S. Pub. No. 2012/0096217 (“Son”) (Ex. 1006);
- viii. U.S. Pub. No. 2008/0189485 (“Jung”) (Ex. 1007);
- ix. U.S. Patent No. 9,665,291 (“Lin”) (Ex. 1008);
- x. U.S. Patent 8,621,137 (“Olbrich”) (Ex. 1010);
- xi. Yang Hu et al., Performance Impact and Interplay of SSD Parallelism through Advanced Commands, Allocation Strategy and Data Granularity (“Hu”) (Ex. 1011);
- xii. U.S. Patent 9,239,781 (“Ellis”) (Ex. 1012);
- xiii. Certified Translation of JP2008176606A (“Takeo”) (Ex. 1013);
- xiv. JP2008176606A (Ex. 1014);
- xv. International Publication No. WO 2008/082996 (“Sinclair-II”) (Ex. 1015);
- xvi. U.S. Pub. No. 2012/0284587 (“Yu”) (Ex. 1016);
- xvii. U.S. Patent 9,423,970 (“Darragh”) (Ex. 1017);
- xviii. Certified Translation of JP2005018490A (“Terasaki”) (Ex. 1018);

Board as Petitioner’s Exhibit 1004. A copy of Sinclair is being filed concurrently with this declaration as Patent Owner’s Exhibit 2019.

xix. JP2005018490A (Ex. 1019).

xx. The exhibits and other documents cited herein.

IV. APPLICABLE LEGAL STANDARDS

A. Level Of Ordinary Skill In The Art

13. My opinions in this declaration are based on the understandings of a person of ordinary skill in the art, which I understand is sometimes referred to as an “ordinary artisan,” or by the acronyms “POSA,” “POSITA” or “PHOSITA,” as of the time of the invention. I understand that the ’801 patent issued on August 29, 2023, and that it claims priority to Provisional Application No. 61/757,464, which was filed on January 28, 2013.

14. I have been told that the person of ordinary skill in the art is a hypothetical person who is presumed to have known the relevant art at the time of the invention. I am instructed that “relevant” means relevant to the challenged claims of the ’801 patent.

15. I understand that factual indicators of the level of ordinary skill in the art include the various prior art approaches employed, the types of problems encountered in the art, the rapidity with which innovations are made, the sophistication of the technology involved, and the educational background of those actively working in the field. I understand that, in assessing the level of skill of a person of ordinary skill in the art, one should consider the type of problems

encountered in the art, the prior solutions to those problems found in the prior art references, the rapidity with which innovations are made, the sophistication of the technology, the level of education of active workers in the field, and my own experience working with those of skill in the art at the time of the invention.

16. In this case, Dr. Litz has asserted in his declaration that a person of ordinary skill in the art as of the time of the '801 patent would have had “at least a Bachelor of Science degree in electrical engineering, or an equivalent field, and at least three years of experience with SSDs or other flash memory devices.” Ex. 1003 [Litz-Decl.] ¶ 43. Petitioner adds to this description an assertion that “[m]ore education can supplement practical experience and vice versa.” Pet. 9. For the purposes of this declaration, I accept Dr. Litz’s proposed qualifications of a POSITA, as qualified by Petitioner. I reserve the right to revisit the issue of the level of knowledge of the POSITA should the Petition be instituted.

17. As further discussed below, my opinions as stated in this declaration are valid even if the Board adopts a somewhat higher or somewhat lower level of ordinary skill in the art than proposed. For example, as will be discussed throughout my report, even a person with the level of knowledge or experience described by Dr. Litz or adopted by the Board as described above would not have a reasonable expectation of success in implementing certain aspects of the proposed combination as of the priority date of the '801 patent.

18. In this report, the absence of an expressed opinion about any section or paragraph of the Petition or Dr. Litz's declaration does not indicate my agreement with the section or paragraph.

B. My Understanding Of Legal Standards

19. When considering the '801 patent and stating my opinions, I rely on the following legal standards as described to me by the attorneys for Radian.

20. I understand that a patent claim is unpatentable if the claimed invention was anticipated by a prior art reference or would have been obvious to a person of ordinary skill in the art at the time of the purported invention.

21. I understand that anticipation requires that every limitation of the claim at issue be disclosed, either expressly or under principles of inherency, in a single prior art reference.

22. I understand that inherency may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient. Instead, it must be proven that the claim limitation not expressly found in the reference necessarily follows from what is found there.

23. I understand that an obviousness analysis involves comparing a claim to the prior art to determine whether the claimed invention would have been obvious to a person of ordinary skill in the art at the time of the invention in view of the prior art and in light of the general knowledge in the art as a whole. I also understand that

obviousness is ultimately a legal conclusion based on underlying facts of four general types, all of which must be considered: (1) the scope and content of the prior art; (2) the level of ordinary skill in the art; (3) the differences between the claimed invention and the prior art; and (4) any objective indicia of non-obviousness.

24. I also understand that obviousness may be established under certain circumstances by combining or modifying the teachings of the prior art. Specific teachings, suggestions, or motivations to combine any first prior art reference with a second prior art reference can be explicit or implicit, but must have existed before the date of purported invention. I understand that prior art references themselves may be one source of a specific teaching or suggestion to combine features of the prior art, but that such suggestions or motivations to combine art may come from the knowledge that a person of ordinary skill in the art would have had.

25. I understand that a reference may be relied upon for all that it teaches, including uses beyond its primary purpose, but also including teachings that lead away from the invention. I understand that a reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, although the mere disclosure of alternative designs does not teach away.

26. I further understand that whether there is a reasonable expectation of success in combining references in a particular way is also relevant to the analysis.

27. I understand that it is improper to use hindsight to combine references or elements of references to reconstruct the invention using the claims as a guide. My analysis of the prior art is made from the perspective of a person of ordinary skill in the art at the time of the invention.

28. I am not offering any legal opinions in this declaration nor am I qualified to do so. I only consider such legal standards in framing my opinions and conclusions as well as placing assertions made by Petitioner in the Petition into the proper context. Additionally, from a subject matter perspective, I understand that the petitioner always has the burden of persuasion regarding a challenge of patentability of an invention under an inter partes review.

V. CLAIMS AND GROUNDS AT ISSUE

29. I understand that Petitioner has challenged the claims of the '801 patent as follows (Pet., 9):

- Claims 1-4, 6, 9-16, 21, 24 based on obviousness over Sinclair and Yamada (Ground I);
- Claims 1, 9, 13-15 based on obviousness over Terasaki and Yamada (Ground II);
- Claims 22 based on obviousness over Sinclair, Yamada, and Olbrich (Ground III);

- Claim 23 based on obviousness over Sinclair, Yamada, Sinclair-II and Yu (Ground IV).

VI. OPINIONS

A. Petitioner's Sinclair Combinations

30. I have reviewed Petitioner's Grounds I, III, and IV, which rely upon Sinclair in combination with Yamada (and in the case of Grounds III and IV, also other references). Pet., 12-65, 91-95.

31. Petitioner's Grounds I, III, and IV rely on U.S. Publication No. 2009/0271562 A1 ("Sinclair"). *See, e.g.*, Pet., iii. All of Petitioner's Sinclair grounds depend upon Petitioner's Sinclair-Yamada Ground I challenge to claim 1.² Accordingly, I have focused my analysis of Petitioner's Grounds I, III, and IV on Petitioner's Ground I argument against claim 1. I reserve the right to supplement my opinions if the Petition is instituted.

² The '801 patent has only one independent claim, and Petitioner's Sinclair-based arguments regarding dependent claims incorporate Petitioner's Ground I challenge to claim 1. *See, e.g.*, Pet., 38-65 (Ground I challenges to dependent claims), 92 (Ground III), 94 (Ground IV).

1. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.

a) *Petitioner Does Not Say What Its Combination Of Sinclair And Yamada Actually Is.*

32. Petitioner discusses its combination of Sinclair and Yamada in the context of a limitation about a memory controller storing a value relating to a “maintenance request issued by [a] host.” *See generally* Pet. 29-30. Petitioner identifies Sinclair’s memory controller as the claimed “memory controller,” Pet., 13-14,³ the initiation of a “flush operation” in Sinclair as the claimed “maintenance request,” *see, e.g.*, Pet., 25,⁴ and Sinclair’s host as the claimed “host,” *see* Pet., 27, and Petitioner is trying to argue that Sinclair’s host issues (or obviously could issue)

³ Sinclair uses the terms “controller,” “system controller,” “memory system controller,” and “memory controller” interchangeably. *See, e.g.*, Ex. 2019 [Sinclair], Fig. 1 (featuring a “memory controller” 108), Fig. 2 (with component labeled 108), [0009] (referring to Figure 2’s “flash memory system controller”), [0044] (referring to “the controller 108” of Figure 1), [0045] (referring to the “system controller 108” of Figure 2).

⁴ Petitioner uses the term “*maintenance* flush operation,” *see* Pet. 27-28, presumably to bolster its identification with the claimed “maintenance request.” Sinclair uses the term “flush operation.”

what it has identified as a “maintenance request.” In connection with this argument, Petitioner discusses aspects of Yamada’s Figure 1, and states that “[i]n the combined Sinclair-Yamada system, the host would issue the flush operation.” *See* Pet., 28-30.

33. Petitioner’s discussion does not identify what “the combined Sinclair-Yamada system” is. In particular, while Petitioner makes passing mention of Yamada components (e.g., “maintenance determination unit,” “command generation unit,” “maintenance entry list”), and while Petitioner includes a highlighted figure showing these components within Yamada’s “access module,” Petitioner does not describe how or whether these components are present in its combination.

34. Yamada discloses, for example, that its “maintenance determination unit” operates on input from a component in Yamada that Petitioner does not discuss in connection with claim 1—a “maintenance request unit” within Yamada’s memory controller (on the right hand side of Yamada’s Figure 1), which populates the “maintenance entry list” that Petitioner mentions. *See, e.g.*, Ex. 1005 [Yamada] [0089] (“The maintenance determination unit 152 stores a logical address included in *each maintenance request transmitted from the maintenance request unit* 115 in a maintenance entry list 153.”); [0100]-[0106] (describing a “Data Maintenance Command Generation Cycle” that begins with input from Yamada’s “maintenance request unit”); [0099] (describing when “the maintenance request unit 115 transmits a maintenance request”).

35. It is unclear if Petitioner proposes that Yamada’s “maintenance determination unit” and “command generation unit” are *part* of an alleged “host” in “the combined Sinclair-Yamada system,” or if Petitioner is only arguing that Yamada teaches, via its disclosures about these components, that a POSITA would modify Sinclair so that Sinclair’s modified “host” issues what Petitioner identifies as “maintenance requests.”

36. In the first case, Petitioner does not state how Yamada’s “maintenance determination unit” would receive information about what alleged “maintenance requests” to issue. In particular, Petitioner does not state that it does this as it does in Yamada—that is, from a copy of Yamada’s “maintenance request unit” within the “controller” of the combined system. *See* Pet., 29. Indeed, Petitioner does not mention any modification to Sinclair’s controller as part of its “combined Sinclair-Yamada system.” *See* Pet. , 29-30 (discussing host-side aspects of Yamada).

37. In the second case, where Petitioner proposes to modify Sinclair in a way inspired by Yamada that does not incorporate Yamada’s components, Petitioner does not identify what component in Sinclair’s host issues alleged “maintenance requests,” how that component gets information about which alleged “maintenance requests” to issue, or on what basis the host issues such requests. Petitioner argues elsewhere that alleged “maintenance requests” in Sinclair *already* come from Sinclair’s host, *see* Pet. 27-28 (referring to what Petitioner terms the “host-issued

flush operation embodiment”). In view of Petitioner’s arguments about unmodified Sinclair, it is not clear how Petitioner’s Sinclair-Yamada combination would differ from how Petitioner argues that unmodified Sinclair already operates.

38. Because Petitioner fails to identify its “combined Sinclair-Yamada system,” Petitioner does not demonstrate that a POSITA would be motivated to arrive at that system, let alone do so with a reasonable expectation of success.

b) *Petitioner Combines Mutually Inconsistent Embodiments Of Sinclair Without Explanation.*

39. Petitioner’s Sinclair grounds rely upon what Sinclair refers to as its “storage address re-mapping” (or “STAR”) “functionality.” Ex. 2019 [Sinclair] [0065]; *see generally, e.g.,* Pet., 25-31 (relying on a “flush operation” performed during Sinclair’s “storage address re-mapping algorithm” for a claim limitation), 27 (referring to the “storage address re-mapping functionality” of Sinclair’s paragraph 65 as “re-mapping (STAR) functionality”). Sinclair describes this functionality in two different types of systems: embodiments where “the storage address re-mapping (STAR) functionality” is “contained *totally within the memory system*,” and an embodiment (Sinclair’s Figure 12) in which “the storage address re-mapping functionality is contained *totally within the host*.” Ex. 2019 [Sinclair], [0059]. Sinclair’s Figures 10 and 11 depict examples of the first type of system, in which the

“STAR” functionality is contained “totally within” a “memory system” (labeled 1004 in Figure 10 and 1102 in Figure 11),

Fig. 10

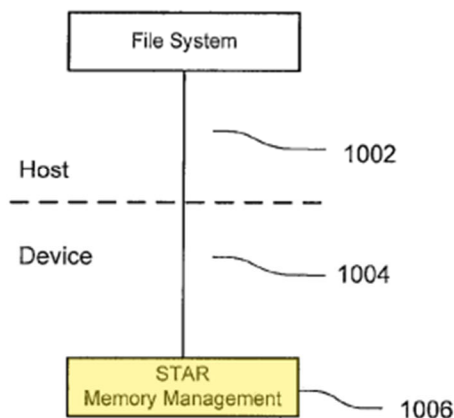
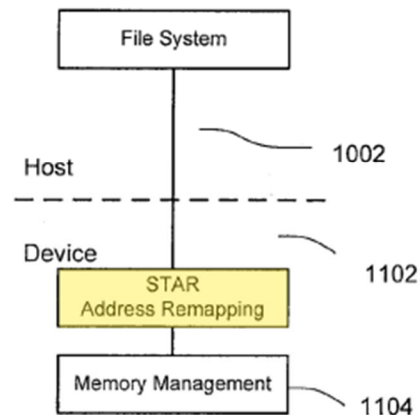
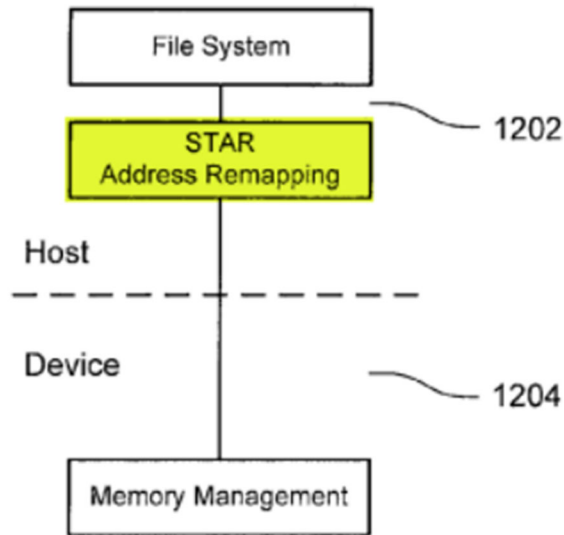


Fig. 11



Ex. 2019 [Sinclair], Figs. 10-11 (highlighted by Patent Owner), [0059], while Sinclair’s Figure 12 is Sinclair’s system in which the “STAR” functionality is contained “totally within” Sinclair’s “host” 1202:

Fig. 12



Ex. 2019 [Sinclair], Fig. 12 (as highlighted by Petitioner, Pet., 28), [0059]. Petitioner’s obviousness argument alternates between Sinclair’s Figures 10-11 and Figure 12 embodiments without explanation.

40. Claim 1 recites, for example, a “host” that Petitioner identifies with Sinclair’s host, *see, e.g.*, Pet. 27, and in connection with claim 1’s recitation of a “maintenance request issued by the host,” Petitioner identifies the claimed “maintenance request” with “the storage address re-mapping algorithm ... initiat[ing] a flush operation” in Sinclair, and expressly relies upon Figure 12’s placement of Sinclair’s STAR functionality to argue that this alleged “maintenance command” is “issued by the host.” *Id.*, 27-28; *see also, e.g., id.* 27 (expressly arguing that “[i]n some embodiments, ‘re-mapping (STAR) functionality’ may be contained

‘within the host’”), 28 (including an annotated copy of Figure 12 and referring to it as “the host-issued flush operation embodiment” of Sinclair); *see also id.*, 30 (citing Sinclair’s Figure 12 as alleged motivation “to combine the teachings of Sinclair and Yamada”), 31 (citing Sinclair’s Figure 12 as alleged evidence of a reasonable expectation of success “in combining Sinclair and Yamada”). Petitioner does not argue that Sinclair’s Figure 11 embodiment discloses this limitation. A POSITA would understand that in Sinclair’s Figure 11 embodiment, what Petitioner identifies as the claimed “maintenance request”—*i.e.*, the initiation of a “flush operation” by Sinclair’s “storage address re-mapping algorithm”—occurs entirely within the controller, and does not involve any request issued by Sinclair’s host. *See, e.g.*, Ex. 2019 [Sinclair] [0077] (describing the beginning of the “block flush process” by stating that “[t]he storage address re-mapping algorithm, ***executed by the controller*** 108 independently for each bank 107A-107D ***in the implementation of FIG. 11,***” designates a block to be flushed).

41. Elsewhere, in connection with claim 1’s recitation of “provid[ing] to the host, responsive to a query received from the host ... the stored value,” Petitioner identifies the claimed “stored value” with a “list of pink blocks that are to be relocated,” Pet., 32, and exclusively relies on the location of STAR functionality in Sinclair’s Figure 12 embodiment—which it terms “the host-issued flush operation embodiment”—to argue that this “stored value” is “provided to the host.” *Id.*, 32;

see also id. (arguing that “for ***the host*** to perform the flush operation, it ***must be provided with*** ... the list of pink blocks that are to be relocated”). Petitioner does not argue that Sinclair’s Figure 11 discloses this limitation. A POSITA would recognize that in Sinclair’s Figure 11, what Petitioner identifies as the claimed “stored value” is not provided to Sinclair’s host.

42. Having thus relied on Sinclair’s Figure 12 for two limitations, Petitioner switches to Figure 11 for other aspects of claim 1.

43. As one example, claim 1 recites a “storage device” comprising “a ***memory controller*** having logic to cause the ***memory controller*** to ... receive write requests ... and, with respect to each one of the write requests, ***identify a specific one of the respective subdivisions*** dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests.” Petitioner identifies the claimed “memory controller” with Sinclair’s memory controller, *see, e.g.*, Pet., 13-14, and identifies the claimed “identify[ing] a specific one of the ... subdivisions” with “identifying a currently open write megablock” in Sinclair. *Id.*, 17-19. The process of identifying a currently open write megablock in Sinclair involves the conversion of “host LBA addresses” to “DLBA addresses.” *Id.* In arguing that this process is performed by Sinclair’s “controller” (as its identifications with elements of claim 1 would require), Petitioner expressly and exclusively relies upon disclosures from Sinclair’s paragraphs 70-72, which describe

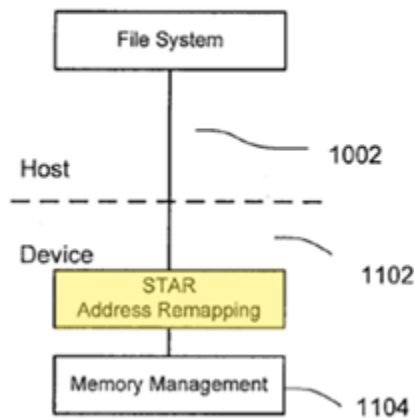
the mapping of “host LBA addresses” to “DLBA addresses” in Sinclair’s Figure 11 embodiment. *See, e.g., id.*, 18-20 (arguing that “Sinclair also discloses that its **controller** identifies the open write megablock,” and extensively quoting from paragraphs 70-72). Sinclair’s paragraph 70 expressly states:

In this example, it is assumed that ***the system configuration of FIG. 11*** is being used, where the host LBA addresses are translated to an intermediate storage LBA address, also referred to as a DLBA address, ***in an application run by the controller 108*** in the memory system 102.

Ex. 2019 [Sinclair] [0070]; *see also, e.g., id.* [0071] (referring to “the write algorithm managed by the controller 108”); Pet. 18-19 (expressly quoting the “in an application run by the controller” language of paragraph 70 while arguing that Sinclair’s controller “identifies the open write megablock”).

44. Sinclair’s Figure 11 embodiment is an embodiment where “the storage address re-mapping (STAR) functionality” (which includes the mapping functions relied on by Petitioner) is contained “totally within” Sinclair’s “memory system” 1102. Ex. 2019 [Sinclair] [0059].

Fig. 11



Ex. 2019 [Sinclair] Fig. 11 (highlighted by Patent Owner).

45. This difference from Sinclair’s Figure 12 embodiment is *exactly* what Petitioner is relying upon to argue that the claimed “identify[ing] a specific one of the respective subdivisions” is performed by what Petitioner has identified as the claimed “controller.” Sinclair’s “controller” is part of Sinclair’s “memory system.” See, e.g., Ex. 2019 [Sinclair] [0070] (specifically referring to “the *controller* 108 *in the memory system* 102” in connection with the Figure 11 embodiment), Fig. 1 (generally locating the “controller” within the “memory system” in embodiments of Sinclair). Sinclair expressly discusses the difference between Figures 11 and 12 in these terms, stating:

The implementation of FIG. 12 would *move* the functionality of storage address re-mapping from the storage device 1204 *to an application on the host 1202*. In this implementation, the function of mapping LBA addresses to DLBA addresses would be similar to that

of FIG. 11, with *the primary difference being that the translation would occur on the host 1202 and not in the memory device 1204.*

Ex. 2019 [Sinclair] [0061]. Here, Sinclair is using the term “memory device” synonymously with what it elsewhere calls a “memory system.” *See, e.g.,* Ex. [0087] (referring to “the memory system 1204” in Figure 12).

46. Sinclair does not discuss hybrid systems in which aspects of its “STAR” functionality are present in both its “host” and its “memory system.” Petitioner is effectively inventing such an embodiment by citing the mutually incompatible Figure 11 and Figure 12 embodiments for different claim limitations, but Petitioner does not explain how it proposes that a POSITA would reconcile the differences between these embodiments.

47. Thus, in my opinion, Petitioner and Dr. Litz fail to show that a POSITA would have been motivated to combine Sinclair and Yamada in the way proposed by Petitioner with a reasonable expectation of success.

2. Petitioner Does Not Show That Its Sinclair Combinations Teach “a maintenance request issued by the host.”

48. Claim 1 recites a “memory controller” having logic relating to “fulfilling a *maintenance request issued by the host.*” Petitioner identifies

Sinclair's memory controller as the claimed "memory controller," Pet., 13-14,⁵ the alleged "initiat[ion]" of a "flush operation" in Sinclair as the claimed "maintenance request," *see, e.g., id.*, 25,⁶ and Sinclair's host as the claimed "host." *See id.*, 27. Petitioner does not show that Sinclair's host "issue[s]" what it identifies with the claimed "maintenance request."

a) *Petitioner Does Not Show Any "maintenance request issued by a host" In Sinclair.*

49. Having made the identifications just mentioned—in particular, having vaguely associated an alleged "maintenance request" with a "flush operation [being] initiated," *id.*, 25—Petitioner states that Sinclair's "flush operations" are themselves carried out at Sinclair's controller. Petitioner quotes Sinclair to the effect that "when

⁵ Sinclair uses the terms "controller," "system controller," and "memory system controller" interchangeably for its "memory controller." *See, e.g.,* Ex. 2019 [Sinclair], Fig. 1 (featuring a "memory controller" 108), Fig. 2 (depicting a component labeled 108), [0009] (referring to Figure 2 as a "flash memory system controller"), [0044] (referring to "the controller 108" of Figure 1), [0045] (referring to the "system controller 108" of Figure 2).

⁶ Petitioner uses the term "***maintenance*** flush operation," *see* Pet. 27-28, but Sinclair uses the term "flush operation." *See supra* note 4.

the storage address re-mapping algorithm detects a need for more white blocks in the bank, the algorithm initiates a flush operation.” *Id.*, 25 (quoting Ex. 2019 [Sinclair] [0065]). Petitioner states that during the flush operation, “*the memory controller* determines and stores the number of pink blocks that need to be relocated.” *Id.*, 25. Petitioner argues that Sinclair “discloses that the flush operation may be *initiated* by the host.” *Id.*, 27.

50. It is not clear how Petitioner’s proposal that an operation “may be initiated by the host” is intended to relate to the claim language. The claim refers to a controller fulfilling a “*maintenance request* issued by the host.” Petitioner’s argument, which focuses on general circumstances under which flush operations might take place, does not identify any purported “maintenance request” in Sinclair.

51. Petitioner argues that “Sinclair discloses that its flush operation performed in fulfilling a maintenance request may be initiated by a host activity, when a host ‘flush cache’ command occurs, or in response to host commands.” *Id.* (internal citations omitted) (citing Ex. 2019 [Sinclair] [0112]). The cited portion of Sinclair does not identify any “maintenance request.” It states:

Preferably, the flush operation acts as a background operation, and thus functions only while the SSD or other portable flash memory device is idle so that host data write speeds are not affected. Alternatively, the flush operation may be utilized in a foreground operation that is active when the host is writing data. If *flush*

operations are arranged as foreground operations, these operations may be automatically suspended when host activity occurs or when a “flush cache” command signifies potential power-down of the SSD or portable flash memory device.

Ex. 2019 [Sinclair] [0112].

52. In saying that Sinclair’s “flush operations” may be “automatically suspended when host activity occurs,” or when a “flush cache” command occurs, Sinclair is not saying that its flush operations are “initiated by a host activity,” or “initiated ... when a host ‘flush cache’ command occurs, or in response to host commands,” as Petitioner would have it. *See* Pet., 27. Sinclair is saying that host activity and host “flush caches” in Sinclair may “*automatically suspend[]*” the operation of Sinclair’s memory controller, such that “flush operations” that might otherwise have taken place at a given time do *not* take place.

53. Petitioner quotes the same disclosure in Sinclair to the effect that a “flush operation may be utilized in a foreground operation that is active when the host is writing data.” Pet., 27 (quoting Ex. 2019 [Sinclair] [0112]). But to say that a flush operation “may be utilized” in an operation “that is active when the host is writing data” is not to say that Sinclair’s host *requested* that flush operation—and does not identify any candidate “maintenance request issued by the host” in Sinclair.

54. Petitioner continues:

A POSITA would understand these disclosures to teach that during foreground operation, the flush operation ... is initiated by a “host activity” such as an activity which requires more white blocks to be created, or at minimum suggest that the maintenance request can be issued by the host.

Pet. 27. But this argument does not identify any “maintenance request issued by the host.”

55. Petitioner also cites Sinclair’s Figure 12 embodiment. Quoting Sinclair, Petitioner observes that Figure 12 depicts an embodiment for which “‘re-mapping (STAR) functionality’ may be contained ‘within the host.’” *Id.* (quoting Ex. 2019 [Sinclair] [0068]). Characterizing Sinclair’s Figure 12 as “the ***host-issued flush operation embodiment***,” Petitioner observes that Sinclair discloses “the ‘host [] may select pink blocks’ for the flush operation” in embodiments of Sinclair’s Figure 12. Pet., 28 (quoting Ex. 2019 [Sinclair] [0068]). Petitioner states that a POSITA would understand the host’s potential involvement in the selection of “pink blocks” (which serve as inputs to Sinclair’s flush operations, *see, e.g.*, Pet., 26) implies or at least “suggest[s] that” commands to perform flush operations themselves “can be issued by the host to the memory controller in various embodiments.” *Id.*, 28.

56. When Petitioner argues that Sinclair’s Figure 12 embodiment “suggest[s] that the maintenance request ***can be*** issued by the host to the memory

controller in *various embodiments*,” *Id.*, Petitioner appears to acknowledge that Sinclair’s system does not require what it identifies as a “maintenance request” to be issued by its host. Indeed, a POSITA would understand that it would be consistent with Sinclair for flush operations to be initiated without the issuance of any request from Sinclair’s host for that purpose.

b) *Petitioner Does Not Show That Yamada Teaches Any “maintenance request issued by the host” In Its Combination.*

57. Petitioner also argues that Yamada teaches a “maintenance request issued by the host.” Pet., 28-30 (arguing that “Yamada discloses moving maintenance operations from the storage controller to the host”). But Petitioner does not relate its identification of “maintenance operations” in Yamada to its identification of other claim elements in Sinclair, or to elements of its combined system. Petitioner states:

Yamada discloses that a maintenance determination unit in the access module 130 may include a “command generation unit” 154 that “generated a maintenance command for performing maintenance for the address of the nonvolatile memory device selected by the maintenance determination unit” 152.

Id., 29.

58. Here, Petitioner identifies a “maintenance request from a host” *in Yamada*: a “maintenance command” in Yamada, as generated by Yamada’s

“command generation unit,” within Yamada’s “access module 130.” But Petitioner does not state that in its modified system, Sinclair’s host is modified to contain a copy of Yamada’s “command generation unit” in its combined system. *See id.* Nor does Petitioner explain how Sinclair’s host would be modified to *implement the functionality* of Yamada’s “maintenance determination unit” and “command generation unit” if it does *not* contain these components. *See id.* Petitioner simply asserts that “[i]n the combined Sinclair-Yamada system, the host would issue the flush operation [**maintenance request issued by the host**] for the pink blocks that need to be relocated [**physically erased**].” *Id.*, 30 (bracketed insertion and boldface in original).

59. I do not agree that Petitioner and Dr. Litz have shown that this limitation is taught, for example, because Petitioner has not explained the impact of Yamada’s alleged teaching of a “maintenance request from a host” in its own system upon the functioning of its combination of Sinclair and Yamada.

3. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller Having Logic To “identify a specific one of the respective subdivisions”

60. Claim 1 recites a “storage device” comprising “a *memory controller* having logic to cause the *memory controller* to ... receive write requests ... and, with respect to each one of the write requests, *identify a specific one of the respective subdivisions* dependent on which one of the non-overlapping ranges

encompasses the address accompanying the one of the write requests.” Petitioner identifies the claimed “memory controller” with Sinclair’s memory controller, *see, e.g.,* Pet., 13-14, the claimed “identify[ing] a specific one of the ... subdivisions” with “identifying a currently open write megablock” in Sinclair, and the claimed “address accompanying the one of the write requests” with “host LBA addresses” provided by Sinclair’s host. *Id.*, 18-19. Petitioner does not involve Yamada in its arguments about this limitation. *See id.*

61. Petitioner does not show how Sinclair uses any host LBA addresses to identify the “currently open write megablock” in Sinclair—let alone that Sinclair performs any such identification “dependent on which one of the non-overlapping ranges encompasses” any host LBA address, as the claim would require under Petitioner’s identifications.

62. Petitioner recognizes that Sinclair’s write algorithm maps incoming host data to “the currently open megapage in the currently open write megablock,” and keeps writing there until the currently open write megablock is full. *See* Ex. 2019 [Sinclair] [0063]; *see also* Pet. 17-18 (quoting Sinclair’s paragraph 63). When a currently open write megablock is full, Sinclair states that “a new write block is allocated from a white block list.” Ex. 2019 [Sinclair] [0064].

63. Sinclair does not state that any LBA addresses provided by its host play any role in this process, and a POSITA would not understand Sinclair's LBA addresses to play any role in this process.

64. Petitioner asserts without support that the controller "identifies the open write megablock ... dependent on which one of the intermediate storage LBA addresses ... encompasses the host LBA addresses." Pet. 18. But the cited text shows no use of either a host LBA or a DLBA to select or otherwise identify which megablock is open for writing; Sinclair describes always writing to whatever megablock is already open. Petitioner's subsequent citation to Sinclair's Figures 16–17 and paragraph 70 shows only that during writing, the open write megablock is "associated with" an LBA run, and that Sinclair's storage address table ("SAT") maps LBA runs to DLBA runs while data is written until the megablock is fully programmed. This does not show the use of any LBA or DLBA in identifying a megablock to write to.

65. Petitioner's argument about this limitation also expressly relies upon Sinclair's paragraphs 70-72, which describe the mapping of "host LBA addresses" to "DLBA addresses" in Sinclair's Figure 11 embodiment. *See, e.g.*, Pet. 18-20 (citing and quoting Ex. 1004 [Sinclair] [0070]-[0072]). As noted above, Petitioner elsewhere exclusively relies upon Sinclair's **Figure 12** embodiment for other aspects of claim 1, and Petitioner's arguments regarding those aspects do not apply to

Sinclair’s Figure 11 embodiment. *See* Section VI.A.1.b) *supra* (describing Petitioner’s reliance on Sinclair’s Figure 12 for “provid[ing] to the host, responsive to a query received from the host ... the stored value”). Similarly, in Sinclair’s Figure 12 embodiment, what Sinclair terms its “storage address re-mapping (STAR) functionality”—which includes the mapping of “host LBA” addresses to “DLBA addresses” relied upon here—is “contained *totally within* the host,” and thus does not evidence a *controller* with logic to perform this functionality. *See* Ex. 2019 [Sinclair] [0059]; *see also id.*, [0061] (stating that “the primary difference” in “mapping LBA addresses to DLBA addresses” between Figure 12 and Figure 11 “being that the translation [in Figure 12] would occur on the host [] and not in the memory device”).

66. Even if one takes Petitioner’s identifications of Sinclair with claim elements for granted for purposes of argument, Petitioner at most argues that some portions of claim 1 correspond with Sinclair’s Figure 11, and that other portions of claim 1 correspond with Sinclair’s Figure 12. Because of the differences between these embodiments as discussed above, Petitioner’s argument does not show how the claimed logic to “identify a specific one of the subdivisions . . .” is present in any single system that simultaneously satisfies all limitations of claim 1.

67. Thus, in my opinion, Petitioner and Dr. Litz fail to show how this limitation is present in Petitioner’s combination of Sinclair and Yamada.

4. Petitioner Does Not Show That Its Sinclair Combinations Teach A Memory Controller “provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value.”

68. Claim 1 recites, *inter alia*, a “storage device” comprising a “memory controller” having logic to cause the memory controller to “provide to the host, responsive to a *query received from the host*, the information, as updated, and the stored value.” Petitioner does not show how this limitation is taught in its Sinclair combinations.

69. Petitioner identifies Sinclair’s memory controller with the claimed “memory controller,” *see, e.g.*, Pet. 13-14, Sinclair’s host with the claimed “host,” *see, e.g., id.*, 27, “a list of red, pink and white blocks for a megablock and their valid page count” in Sinclair for the claimed “information, as updated,” and “a listing of pink blocks that are to be relocated” in Sinclair for the claimed “value.” *Id.*, 32.

70. To attempt to identify something in Sinclair as the claimed “query received from the host” that would prompt Sinclair’s controller to provide Sinclair’s host with what Petitioner identifies as the claimed “information, as updated” and “stored value,” Petitioner exclusively relies upon Sinclair’s Figure 12 embodiment. *See id.* 32; *see also id.* 27-28 (identifying Figure 12 as “the host-issued flush operation embodiment”).

71. As noted above, however, Petitioner elsewhere exclusively relies upon Sinclair’s Figure 11 embodiment for allegedly teaching other aspects of claim 1, and Petitioner’s arguments regarding those aspects do not apply to Sinclair’s Figure 12 embodiment. *See* Section VI.A.1.b) *supra*. Similarly, Petitioner does not and cannot argue that Sinclair’s Figure 11 teaches the claimed “query from the host.” Sinclair expressly discloses that in its Figure 11 embodiment, “the storage address re-mapping (STAR) functionality”—which manages and uses the lists of blocks that Petitioner identifies with the claimed “information” and “stored value—is “contained ***totally within*** the memory system.” Ex. 2019 [Sinclair] [0059]. Sinclair says that locating this functionality “totally within” the memory system allows the system to “operate with a ***legacy host*** 1002 with ***no modifications required on the host*** 1002.” *Id.*

72. A POSITA would understand from this and other disclosures in Sinclair that in the Figure 11 embodiment, there is no reason for the host to query the controller for the information identified by Petitioner, and no reason for the memory controller to be configured to provide that information to the host.

73. In failing to explain how Sinclair’s distinct embodiments apply to claim 1, Petitioner thus fails to show how the claimed “query from the host” is present in its combined system. Even if one takes Petitioner’s identifications of Sinclair with claim elements for granted for purposes of argument, Petitioner at most argues that

some portions of claim 1 correspond with Sinclair's Figure 11, and that other portions of claim 1 correspond with Sinclair's Figure 12. Because of the differences between these embodiments as discussed above, Petitioner's argument does not show how the claimed logic to cause a memory controller to "provide to the host, responsive to a *query received from the host*, the information, as updated, and the stored value" is present in any single system that simultaneously satisfies all of the limitations of claim 1.

74. Petitioner also argues that Yamada teaches the claimed "query received from the host." Petitioner argues that "Yamada discloses a maintenance determination unit with a maintenance entry list to perform maintenance operations" and that "Yamada teaches that the controller maintains information about the valid block count and that this information can be retrieved via host queries." Pet. 32-33 (citing Ex. 1005 [Yamada] [0032], [0096]).

75. The disclosures in Yamada cited by Petitioner do not show that Yamada's "information about the valid block count" is retrieved via host queries. The first paragraph cited by Petitioner refers to an "error correction unit" within Yamada's controller, and a "maintenance request unit" within Yamada's controller "generat[ing] a maintenance request." It reads in full:

The error correction unit determines the number of errors included in data read from the nonvolatile memory, and corrects an

error included in the read data. The maintenance request unit generates a maintenance request indicating that the nonvolatile memory needs maintenance to the access device when the number of errors determined by the error correction unit is greater than or equal to a criterion value.

Ex. 1005 [Yamada] [0032]. This disclosure does not refer to anything in Yamada being retrieved via host queries, but instead concerns operations internal to Yamada's controller.

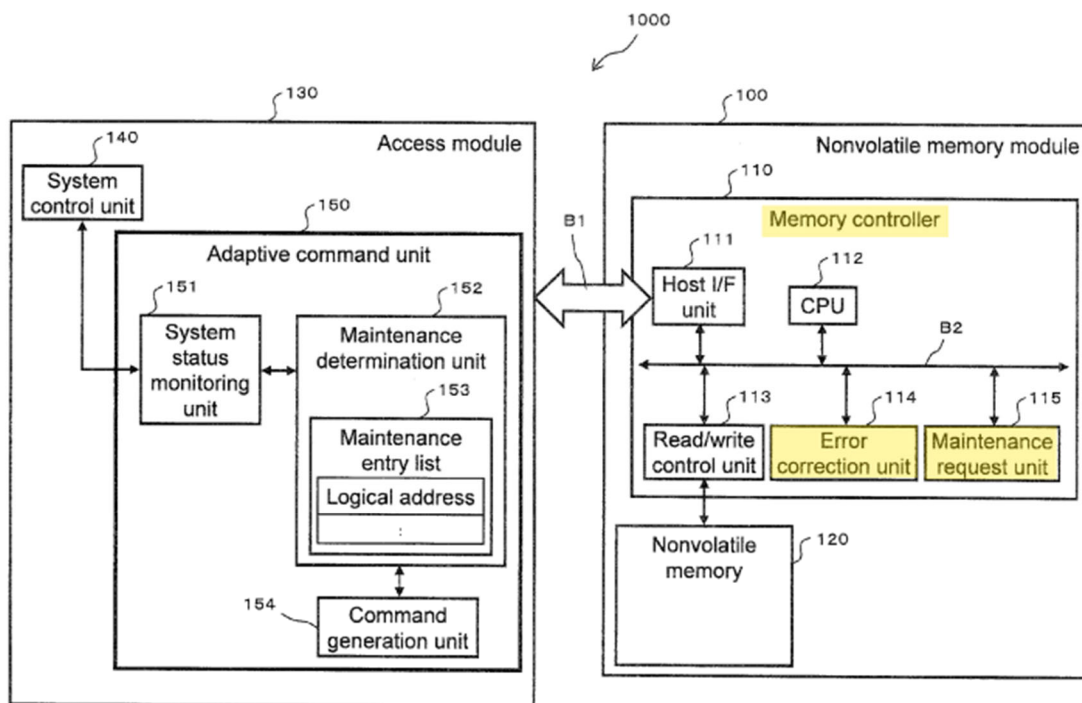


FIG. 1

See Ex. 1005 [Yamada], Fig. 1 (“Error correction unit” and “Maintenance request unit” within the “Memory controller” highlighted by Patent Owner).

76. The second paragraph cited by Petitioner, which also describes aspects of the “error correction unit” within Yamada’s controller, says that a “CPU” on the

controller “*transmits* ... data from the error correction unit” to an “access module” on Yamada’s host:

The error correction unit 114 determines whether the read data has an error, and determines the number of errors in the data. When detecting an error in the data, the error correction unit 114 corrects the erroneous data. After completing the error correction, *the CPU 112 transmits the data from the error correction unit 114 to the access module 130*, and completes the data reading process (S201).

Ex. 1005 [Yamada] [0096].

77. This disclosure does not show any queries issued by what Petitioner refers to as Yamada’s “host”—Yamada’s “access module.”⁷ Yamada’s Figure 1 graphically depicts the locations of the error correction unit and CPU within Yamada’s controller, not Yamada’s “access module” (Petitioner’s “host”)

⁷ While Yamada does not call its “access module” a “host,” Yamada expressly refers to the component of its controller that interfaces with the “access module” as the “host interface unit” or “host I/F unit.” *See, e.g.*, Ex. 1005 [Yamada] [0076]-[0077] & Fig. 1. A POSA would understand Petitioner’s reference to “host queries” in Yamada (Pet., 33) to refer to queries made by Yamada’s access module.

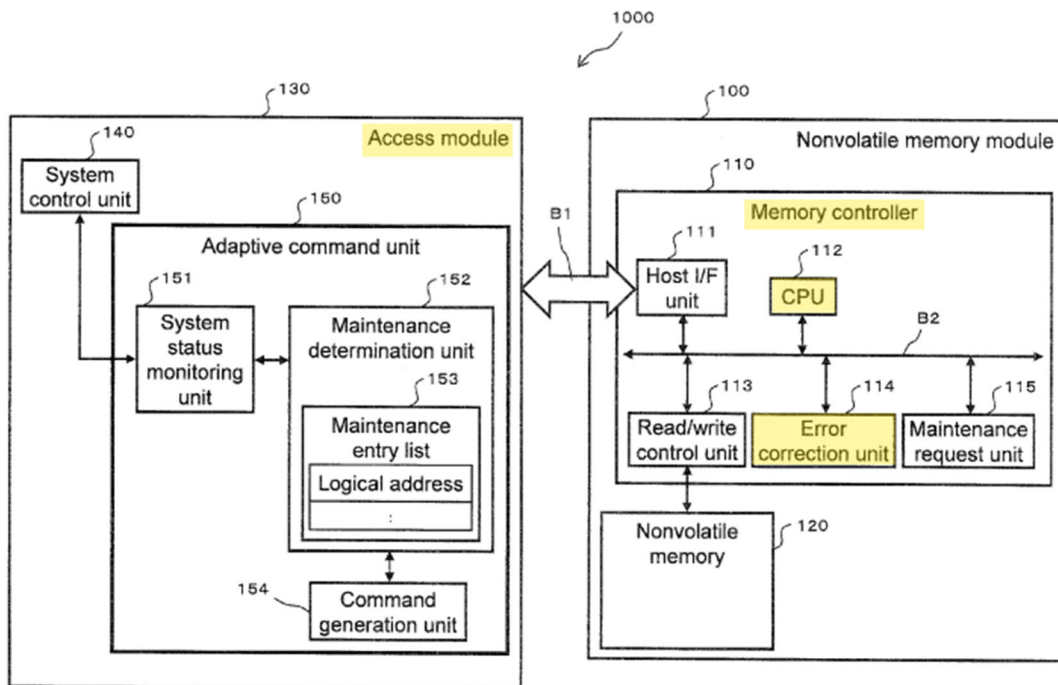


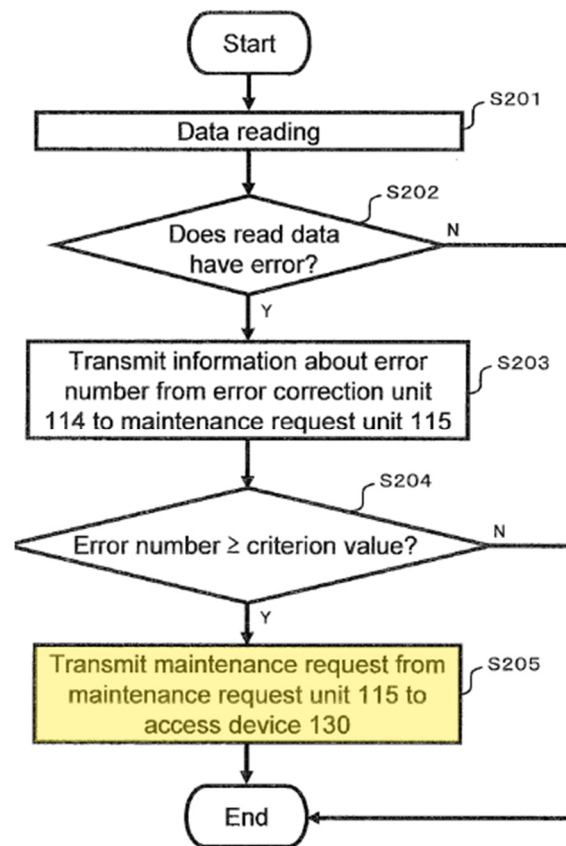
FIG. 1

Ex. 1005 [Yamada], Fig. 1 (highlighted by Patent Owner).

78. When Yamada refers to its “CPU ... transmit[ting] the data from the error correction unit 114 to the access module,” a POSITA would understand this “transmi[ssion]” to be an activity initiated by the CPU within Yamada’s controller. Similarly, in Yamada’s paragraph 32, which refers to how Yamada’s “maintenance request unit generates a maintenance request” after its error correction unit has corrected an error data, a POSITA would understand that this maintenance request arrives at Yamada’s access module by being sent there in activity initiated by the CPU within Yamada’s controller.

79. Indeed, a POSITA would generally understand that this is how maintenance requests in Yamada arrive at Yamada's access module. In the course of generally describing "[t]he internal structure of [its] memory controller," Ex. 1005 [Yamada] [0075]-[0082], Yamada states, for example, that "[w]hen determining that the area needs maintenance, the maintenance request unit 115 transmits a maintenance request to the access module 130," *id.*, [0081], and that "[t]he maintenance request is transmitted to the access module 130 via the host I/F unit 111." *Id.*, [0082]. Both the "maintenance request unit 115" and the "host I/F unit 111" are components within Yamada's memory controller; Yamada also states, for example, that "[t]he CPU 112 controls the overall operation of the memory controller 110." *Id.*, [0078]. Yamada's "maintenance requests" originate at Yamada's controller and are sent to Yamada's "access module" without anything resembling a query from the "access module" to produce them.

80. This understanding of the POSITA is confirmed by other disclosures in Yamada, which uniformly refer to the "transmission" of maintenance requests to Yamada's "access module." Yamada's Figure 2, for example, is a flowchart of the process that Yamada's paragraph 96 (cited by Petitioner), and labels the relevant step "Transmit[ting] maintenance request *from* maintenance request unit 115 *to* access device 130":



Ex. 1005 [Yamada], Fig. 2 & [0096].

81. All of these steps are “performed by [the] nonvolatile memory module” [memory controller] in Sinclair. *Id.* [0039].

82. Disclosures in Yamada relating to the host-side *processing* of controller-provided maintenance requests (what Yamada calls its “Data Maintenance Command Generation Cycle”) are in accord: this processing begins with the receipt of a maintenance request from Yamada’s controller, and Yamada never mentions anything like a “query” that initiates this processing. Yamada generally refers to processing as “[t]he procedure performed by the access module

130 *when receiving a maintenance request* from the nonvolatile memory module 100,” for example, and describes its first step as occurring “[w]hen the access module 130 receives a maintenance request from the nonvolatile memory module 100 (S301).” Ex. 1005 [Yamada] [0100]-[0101]; *see also id.*, Fig. 3 (depicting “Receive maintenance request from nonvolatile memory module 100” as the first step after the “Start” of Yamada’s “Data Maintenance Command Generation Cycle”). This feature is consistent across every embodiment where Yamada describes its “access module” processing a maintenance request based on data provided by Yamada’s controller. *See, e.g., id.* [0131] (stating that in Figure 6, “the access module 500 *first receives a maintenance request* from the nonvolatile memory module 100”), [0156] (similar, for Figure 8).

83. Thus, in my opinion, Petitioner and Dr. Litz fail to show a “storage device” comprising a “memory controller” having logic to cause the memory controller to “provide to the host, responsive to a *query received from the host*, the information, as updated, and the stored value” in Petitioner’s combination of Sinclair and Yamada.

B. Petitioner’s Terasaki Combination

I have reviewed Petitioner’s Ground II, which involves a combination of Terasaki and Yamada. Pet., 66-91. All of Petitioner’s Terasaki-based challenges

depend upon Petitioner's Terasaki-Yamada combination as identified in connection with claim 1.⁸

1. Petitioner Does Not Show A Motivation To Combine With A Reasonable Expectation Of Success.

a) *Petitioner Does Not Say What Its Combination Of Terasaki And Yamada Actually Is.*

84. Petitioner identifies its combination of Terasaki and Yamada in connection with a limitation relating to a “maintenance request issued by [a] host.” Pet., 77-78. At the relevant point in Petitioner's argument, Petitioner has identified what it terms an “erase processing request” in Terasaki as the claimed “maintenance request,” and is arguing that Terasaki teaches that this “erase processing request” comes from (or at least could obviously come from) Terasaki's host system. *Id.*, 77. Petitioner implicitly describes its combination of Terasaki with Yamada as follows:

To the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada. Yamada discloses moving maintenance operations from the storage controller to its “maintenance determination unit” and “command generation unit” at the host for

⁸ The '801 patent has only one independent claim, and Petitioner's arguments for dependent claims all incorporate and rely upon its arguments for claim 1. *See generally* Pet., 85-91 (addressing dependent claims in Ground II).

causing maintenance for data retention, such as refresh, to be performed for an address stored in the maintenance entry list. In the combined Terasaki-Yamada system, the host would issue the erase processing request [**maintenance request issued by the host**] to change zones flagged as “erase requested” to an “erased status.”

Id. 77-78 (bracketed insertion and boldface in original) (citations omitted).

85. This does not identify “the combined Terasaki-Yamada system.” Just as with Petitioner’s Sinclair combinations, it is not clear if Petitioner proposes Yamada’s “maintenance determination unit” and “command generation unit” would be *part* of any alleged “host” in “the combined Terasaki-Yamada system,” or if Petitioner is only arguing that Yamada somehow *teaches*, via its disclosures, that a POSITA would modify Terasaki so that Terasaki’s as-modified “host” issues what Petitioner terms “erase processing requests.”

86. In the first case, Petitioner leaves open how Yamada’s “maintenance determination unit” receives any information about what alleged “maintenance requests” to issue. Petitioner does not, for example, discuss any modification to Terasaki’s controller as part of its “the combined Terasaki-Yamada system.” *See id.*, 78 (Petitioner highlighting only host-side aspects of Yamada’s system).

87. If Petitioner is alternately proposing that Terasaki could be modified in a way inspired by Yamada that does *not* rely on Yamada’s specialized components, Petitioner does not identify what those modifications would be. Petitioner’s

mapping against claim 1 elsewhere identifies certain “write requests from a host,” *id.*, 76, and “requests that change the erase flag to an erased flag ... [as] initiated by the host,” *id.*, 77, as alleged “maintenance requests” in unmodified Terasaki—but does so in connection with arguing that they already come from Terasaki’s host. *See id.* 76-77. In view of Petitioner’s arguments about unmodified Terasaki, it is not clear how Petitioner’s Terasaki-Yamada combination would differ from how Petitioner argues that unmodified Terasaki already operates. Petitioner’s argument that if “maintenance requests” did not already come from Terasaki’s host, “the host would issue the ... request” in “the combined Terasaki-Yamada system,” *id.*, 78, does not identify the system.

88. Because Petitioner and Dr. Litz fail to identify its “combined Terasaki-Yamada system,” in particular, Petitioner and Dr. Litz do not show that a POSITA would be motivated to arrive that that system, let alone do so with a reasonable expectation of success.

2. Petitioner Does Not Show That Its Terasaki Combination Teaches Any “maintenance request issued by the host.”

89. Claim 1 recites a memory controller having logic to, *inter alia*, physically erase memory in fulfilling “a maintenance request issued by the host.” Petitioner does not show how this limitation is met in its combination of Terasaki and Yamada.

- a) *Petitioner Does Not Show Any “maintenance request issued by [a] host” In Terasaki.*

90. Petitioner makes several attempts to identify a “maintenance request issued by [a] host” in Terasaki, but does not show that the limitation is present.

91. First, Petitioner states: “during a subsequent write request from the host when maintenance is needed, an erase process is performed **[fulfilling a maintenance request issued by the host]** where the block is changed to an erased block.” Pet. 76 (bracketed insertion and boldface in original).

92. It is not clear how Petitioner’s juxtaposition of a characterization of Terasaki with claim language is intended to identify an alleged “maintenance request issued by the host.” Terasaki never uses the term “write request.” Terasaki speaks of write “commands.” *See, e.g.*, Ex. 1018 [Terasaki] [0007] (referring to “a write command ... received from the host system”), [0025] (referring to “external commands supplied by [the] host system”), [0069] (referring to “write processing for data ... according to an external command from the host”).

93. Because Petitioner refers to a “write request from the host” in Terasaki, and because claim 1 recites a “maintenance request issued by a host,” one might infer that Petitioner is identifying a write command in Terasaki as the claimed “maintenance request from a host.” But Terasaki’s “write commands” do not show the claimed “maintenance request.”

94. A POSITA would not understand a host-initiated write command in Terasaki's system to be a "maintenance request" in the sense of the '801 patent. Terasaki describes its host-initiated write commands as having two inputs: the data to be written and a range of logical block addresses, *see, e.g.*, Ex. 1018 [Terasaki] [0046] (referring to an exemplary "external command to write data corresponding to a logical block address for LBA15000 to LBA15999 [being] received from the host system"), and a POSITA would understand that such a command in Terasaki is an instruction for Terasaki's controller to simply **store** the corresponding data at the corresponding range of logical block addresses. Such a command is not a request for maintenance.

95. The '801 patent reflects a high-level difference between normal input/output and "maintenance," for example, when discussing its Figure 1B embodiment. There, the '801 patent describes a "host processor" using "host software ... to schedule and manage memory operations, including both **memory transactions** and **memory maintenance** operations." Ex. 1001 ['801] 11:16-20; *see also, e.g., id.*, 28:36-38 (distinguishing "commands for memory transactions" from "configuration commands," "query commands," and other interactions). A POSITA would recognize host-initiated write requests in an embodiment of the '801 patent like Figure 1B to be examples of "memory transactions" that are not "memory maintenance operations."

96. The '801 patent also reflects a high-level difference between normal input/output and “maintenance” in its claim language. In one place, for example, claim 1 recites a controller having logic causing the controller to receive “write requests, accompanying addresses and accompanying data from a host,” while in another place, claim 1 recites “a maintenance request issued by the host.” A POSITA would not understand the claimed “write requests” and the claimed “maintenance request” to be literally the same thing—which is what Petitioner is effectively doing when it identifies host write commands in Terasaki with the claimed “write requests ... issued by the host,” Pet., 69-70, if Petitioner is separately also identifying host write commands in Terasaki with the claimed “maintenance request issued by a host.” *Id.*, 76.

97. Because of Petitioner’s lack of specificity in identifying the alleged “maintenance request issued by the host” it is conceivable that Petitioner does not intend *all* host write commands in Terasaki to be synonymous with the claimed “maintenance request issued by a host.” It is conceivable, for example, that in saying “a subsequent write request from the host when maintenance is needed, an erase process is performed [**fulfilling a maintenance request issued by the host**],” *id.* 76 (bracketed insertions and boldface in original), Petitioner is not identifying an *arbitrary* write command in Terasaki with the claimed “maintenance request issued by a host,” but is instead attempting to identify only a write command in Terasaki

that occurs “when maintenance is needed” in Terasaki as the claimed “maintenance request issued by a host,” or perhaps that only a write command in Terasaki that occurs **both** “when maintenance is needed” **and** which results in an “erase process [being] performed” as the claimed “maintenance request issued by a host.” But these are distinctions without a difference.

98. Terasaki does not itself provide a basis for distinguishing a host-issued write command that arrives at Terasaki’s controller “when maintenance is needed”—a phrase used by Petitioner that does not appear in Terasaki—from any other write command in Terasaki, such that some host write commands would qualify as “maintenance requests” and others would not. Similarly, Terasaki does not provide a basis for distinguishing host write commands that result in erase operations being performed within Terasaki’s controller from host write commands that do not, such that the former would qualify as “maintenance requests” and the latter would not.

99. Terasaki never expressly indicates that its host (whose write commands provide only logical block address ranges and data to be written) is even aware that it is interacting with flash storage as distinguished from any other kind of storage. Terasaki also does not describe any way that Terasaki’s host could determine in advance which of its write commands would result in erase operations being performed by Terasaki’s controller, and which of its write commands would not.

The information needed to make such a determination is present in structures like Terasaki's "zone management table" (*see* Ex. 1018 [Terasaki] [0045]-[0047]), but Terasaki locates these structures on its controller and does not describe its host consulting them as a part of issuing write commands.

100. Petitioner appears to admit that Terasaki does not disclose its host having access to its "zone management table," for example, because in connection with another limitation, Petitioner relies on another reference, Yamada, to create a system in which the host of the combined system is said to *literally* issue "erase processing requests," (that is, not just host write commands that *result in* erase processing requests). Pet., 79. Petitioner argues that in this combined system, "the host would maintain Terasaki's zone management table." *Id.* A POSITA would understand that in unmodified Terasaki, the host does not maintain Terasaki's zone management table.

101. A POSITA would recognize that in (unmodified) Terasaki, literally the exact same host write command—a command, for example, to write specific data to "a logical block address for LBA10000 through LBA10999" (to use an exemplary of LBAs given in Terasaki, *see id.* [0053]-[0056])—might in some instances result in flash being erased by Terasaki's controller, and in other instances not have this effect. In such circumstances a POSITA would *not* understand that some of these write commands were "maintenance requests" in the sense of the '801 patent and

others were not. The structure of Terasaki's host write commands does not provide a basis for any such distinction. A POSITA would not recognize *any* host write command in Terasaki to be a "maintenance request" in the sense of the '801 patent.

102. As a second putative "maintenance request from a host," Petitioner states that "Terasaki also discloses that requests that change the erase request flag to an erased flag [maintenance] may be initiated by the host [request issued by the host]." Pet., 77 (bracketed insertions and boldface in original). The cited support for this assertion is the following a high-level description of the "host interface block" in Terasaki's Figure 1:

Host interface block 7 is a functional block that exchanges data, address information, status information, and external command information with host system 4. In other words, when flash memory system 1 is mounted onto host system 4, flash memory system 1 and host system 4 are connected to each other via external bus 13. In such a state, data and the like that is supplied from host system 4 to flash memory system 1 is taken into the interior of controller 3 using host interface block 7 as an inlet, and data and the like that is supplied from flash memory system 1 to host system 4 is fed into host system 4 using host interface block 7 as an outlet.

In addition, host interface block 7 includes a task file register (not shown in the figure) that temporarily stores host addresses and external commands supplied by host system 4, and an error register (not shown in the figure) that may be set in the event of an error.

Ex. 1018 [Terasaki], ¶¶ 24-25.

103. This disclosure says nothing about Terasaki’s **host** “request[ing] [to] change the erase request flag to an erased flag.” *Cf.* Pet. 77 (making this assertion). In particular, it does not identify any alleged “maintenance request issued by a host.”

104. Petitioner attempts to connect the disclosure to its argument by arguing that because Terasaki discloses “that its controller and host exchange data/addresses and status/command information via host interface block which includes a task file register storing addresses and commands[,] [a] POSITA would understand these disclosures to teach that the necessary erase processing that is performed on zones flags as ‘erase request’ is initiated from command information sent by the host to the controller.” *Id.*

105. This argument does not engage with the claim language and Petitioner’s own mappings of claim 1 onto Terasaki. To summarize, Petitioner has just identified “requests that change the erase request flag to an erased flag” in Terasaki as the claimed “maintenance request[s].” *Id.* Petitioner is now attempting to argue that these requests are “issued by [Terasaki’s] host,” as its mapping would require. *Id.* And instead of showing why the alleged “requests” just identified by Petitioner are, in fact, **issued by** Terasaki’s host, Petitioner says only that a POSITA would allegedly understand that “***the necessary erase processing*** that is performed on zones flags ... is ***initiated from command information sent by the host*** to the

controller.” *Id.* In speaking in terms of “erase processing” being “initiated” from “command information sent by the host” that has not previously been identified with claim limitations, Petitioner is changing the subject—not even stating, let alone showing, that what Petitioner has identified as the claimed “maintenance request[s]” are “issued by [Terasaki’s] host.” This does not show how the limitation is present in Terasaki or in Petitioner’s combination of Terasaki and Yamada.

b) *Petitioner Does Not Show That Yamada Teaches Any “maintenance request issued by the host” In Its Combination.*

106. Petitioner argues “[t]o the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada.” Pet., 77.

107. Petitioner subsequently states that “[i]n the combined Terasaki-Yamada system, the host would issue the erase processing request [**maintenance request issued by the host**] to change zones flagged as ‘erase requested’ to an ‘erased status.’” *Id.*, 78 (bracketed insertion and boldface in original).⁹ With this disclosure, Petitioner presumably identifies a command “to change zones flagged as ‘erased requested’ to an ‘erase status’” in Terasaki—or perhaps a command to erase

⁹ Terasaki does not use the phrase “erased status,” but uses the word “erased” to describe a “status” of a zone. *See, e.g.*, Ex. 1018 [Terasaki], Fig. 8, [0045].

a Terasaki zone that has the *result* of changing a zone flagged as “erase requested” to a zone flagged “erased”—as the claimed “maintenance request issued by the host” in its argument.

108. Petitioner does not link this command in *Terasaki* to any structure in “the combined Terasaki-Yamada system.” Indeed, Petitioner does not identify what “the combined Terasaki-Yamada system” is. *See, e.g.,* Section VI.B.1.a) *supra*. Petitioner’s entire argument about the combination is a single paragraph.

To the extent Terasaki does not teach the erase processing request may be initiated by the host, it would have been obvious in further view of Yamada. Yamada discloses moving maintenance operations from the storage controller to its “maintenance determination unit” and “command generation unit” at the host for causing maintenance for data retention, such as refresh, to be performed for an address stored in the maintenance entry list. In the combined Terasaki-Yamada system, the host would issue the erase processing request [**maintenance request issued by the host**] to change zones flagged as “erase requested” to an “erased status.”

Pet. 77-78 (bracketed insertion and boldface in original) (citations omitted).

109. This partial description of Yamada’s “maintenance determination unit” and “command generation unit” does not say whether “the combined Terasaki-Yamada system” contains *copies* of these components, or whether “the combined Terasaki-Yamada system” is a modified version of Terasaki that adopts the

functionality of these components. Indeed, Petitioner does not describe what the combined system would be, other than to say it would contain the claimed “maintenance request issued by the host.”

110. Thus, in my opinion, Petitioner and Dr. Litz fail to show “a maintenance request issued by the host” in its combination of Terasaki and Yamada.

3. Petitioner Does Not Show That Its Terasaki Combination Teaches Logic To Cause A Memory Controller To “identify a specific one of the respective subdivisions”

111. Claim 1 recites, *inter alia*, a memory controller having logic to cause the memory controller to “receive write requests, accompanying addresses and accompanying data from a host and, with respect to each one of the write requests, [to] identify a specific one of the respective subdivisions dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests.” Petitioner identifies a “write command” in Terasaki, which may include a logical address and data, with the claimed “write request[]” and “accompanying address[] and accompanying data,” Pet. 69-70, and the claimed “subdivisions” with Terasaki’s “zones.” *Id.*, 67-68. But Petitioner does not show how Terasaki’s controller allegedly “identif[ies] a specific one of the respective subdivisions” [a Terasaki zone, in Petitioner’s read] “dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write

requests.” Petitioner does not refer to Yamada in argument about this limitation. *See id.*, 70-71.

112. Petitioner states that “Terasaki discloses that a corresponding LBA indicates which LBA the block corresponds to when data is stored in the block.” *Id.*, 70. Petitioner does not link this disclosure to any identification of “a specific one” of any Terasaki “zones” from an address accompanying a write command.

113. Petitioner then says that “Terasaki discloses a zone management table that manages the storage status of each zone and manages the LBA of the data written onto each zone by zone, as show[n] in Fig. 6.” *Id.*, 70-71. Again, however, Petitioner does not link this table to the identification of “a specific one” of any Terasaki “zone” from any address accompanying a Terasaki write command, let alone show Terasaki doing this “dependent on which one of the non-overlapping ranges encompasses” an address accompanying a write command.

114. Petitioner states that “[t]his initial stored data is written onto the zone management table above for the following LBAs,” and lists an exemplary correspondence between Terasaki “zones” and ranges of LBAs in Terasaki’s Figure 6. *Id.* But again, this does not show the limitation.

115. Terasaki’s Figure 6 depicts a correspondence between exemplary Terasaki “zones” and certain ranges of LBAs in Terasaki, but does not support Petitioner’s contention that Terasaki’s controller somehow uses a host-supplied

address from a write command in Terasaki to identify a Terasaki “zone” to write to. As Petitioner appears to acknowledge, Terasaki describes its controller processing a host command to write to a logical address by using its “zone management table” to find a Terasaki zone in an “erased” status, and writing the data to that Terasaki zone. *See, e.g.*, Ex. 1018 [Terasaki] [0046], Pet., 72 (citing paragraph 46 of Terasaki’s discussion of its Figure 6).

116. Terasaki does not disclose any selection rule by which its controller selects a **particular** “erased” Terasaki zone to write to. As Petitioner elsewhere describes this process, “an empty erased zone is found within the zone management table,” Pet., 72; Terasaki similarly only states that upon a host write request, an “erased zone will be found within the zone management table, and the data will be written onto that erased zone.” Ex. 1018 [Terasaki] [0046]. A POSA would not understand from these disclosures that an address accompanying the host write request is necessarily or even advantageously used in selecting an erased zone to write to. It would be consistent with Terasaki, for example, for the controller to select a zone on a basis entirely unrelated to any address supplied by the host—such as simply selecting the lowest-numbered zone that is in an “erased” status, or selecting an erased zone on the basis of wear information (such as Terasaki’s “erase count ... for each zone,” which Terasaki expressly suggests could be used “to manage the order of priority of the erased zones to be used,” *id.* [0049]). Terasaki gives an

example in which when LBA 11000 is rewritten, it is written to Zone 7 when it was previously in Zone 2—another indication that the zone chosen for writing does not depend on the LBA. *Id.* [0048].

117. Thus, in my opinion, Petitioner and Dr. Litz fail to show logic to cause the memory controller to “receive write requests, accompanying addresses and accompanying data from a host and, with respect to each one of the write requests, [to] identify a specific one of the respective subdivisions dependent on which one of the non-overlapping ranges encompasses the address accompanying the one of the write requests” in Petitioner’s Terasaki combination.

4. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller “stor[ing] a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host.”

118. Claim 1 recites, *inter alia*, a memory controller having logic to cause the memory controller to “store a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host.”

119. Petitioner’s discussion of “a value representing a size of the NAND flash memory that is to be physically erased” does not identify where this limitation is said to appear in Terasaki. In connection with identifying the claimed “value,” Petitioner states:

When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet., 76 (bracketed insertion and boldface in original) (citing Ex. 1003 [Litz-Decl.] ¶ 239) (citations omitted).

120. This argument does not make clear what Petitioner contends is a “value representing a size of the of the NAND flash memory that is to be physically erased.” There are various conceivable candidates: the “1000” in “1000 blocks,” the phrase “up to 1000 blocks” (although Petitioner does not expressly identify which blocks), and the status of the “erase request flag” for the “up to 1000 blocks” (playing the role of a potentially non-numerical “value”), for example. But Petitioner’s argument does not single out one of these things as the claimed “value,” or show how it represents a “size of the NAND flash memory that is to be physically erased,” or show how Terasaki’s controller “store[s]” the claimed “value.”

121. Petitioner cites its expert Dr. Litz, who states:

When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data was written is changed to an erase request flag for the up to 1000 blocks of that one zone, *which represents a size of the NAND flash memory that is to be*

physically erased. EX1018, ¶68. This erase request flag indicates that erase processing for that zone is necessary. *Id.*

Ex. 1003 [Litz-Decl.] ¶ 239. But this paragraph, like the petition itself, does not identify what Petitioner alleges to be the claimed “value.”

122. Petitioner and Petitioner’s expert both cite the following paragraph in Terasaki:

If the data is rewritten, the zone erase process described above will be performed on the zone in which the original data was written. Therefore, if the data is rewritten, it is preferable to change the written flag written on the first page of the initial good block in the zone in which the original data was written to the erase request flag, and then to change the erase request flag to the erased flag when the above-noted zone erase processing has been performed.

Ex. 1018 [Terasaki] [0068]. But this paragraph, like the petition itself, also does not identify what Petitioner alleges to be the claimed “value.” And the knowledge of the POSITA does not resolve this ambiguity.

123. I am not certain of what Petitioner proposes to identify as the “value representing a size of the NAND flash memory that is to be physically erased.” As one example, I do not understand Petitioner’s use of the qualifier “for the up to 1000 blocks of that one zone” in the statement that “the written flag ... in the zone in which the original data is written is changed to an erase request flag for the up to

1000 blocks of that one zone.” Pet., 76. A POSITA would understand from Terasaki’s various express disclosures that the “erase request” flag in Terasaki is set on a zone basis, not on the basis of individual blocks within a zone. *See, e.g.*, Ex. 1018 [Terasaki] [0047] (“when revising data that has already been written, the zone management table ... will change the storage status *of that zone* to an erase request flag (a flag indicating that erase processing is necessary to convert *the zone* to *an erased zone*)”), [0049] (referring to “information indicating the state of each zone (write state, erase state, erase request state, etc.)”). This is depicted graphically, for example, in Terasaki’s Figure 8, which depicts a “Zone 2 in which the original data was written ... changed to an erase request flag.” *Id.* [0048].

(FIGURE 8)

Zone	Logical block address
Zone 1	LBA10000
Zone 2	Erase request
Zone 3	LBA12000
Zone 4	LBA13000
Zone 5	LBA14000
Zone 6	LBA15000
Zone 7	LBA11000
Zone 8	Erased
Zone 9	Erased
Zone 10	Erased
Zone 11	Erased
Zone 12	Erased
Zone 13	Erased
Zone 14	Erased
Zone 15	Erased
Zone 16	Erased
Zone 17	Erased
Zone 18	Erased
Zone 19	Erased
Zone 20	Erased

Id., Fig. 8 (highlighting added). Thus, when Petitioner speaks of changing a flag “to an erase request flag for the up to 1000 blocks of that one zone,” for purposes of identifying the claimed “value representing a size of the NAND flash memory that is to be physically erased,” Petitioner’s meaning is unclear, as the erase request flag, if it is set, is set for the entire zone—not for individual blocks within the zone, and not for “up to 1000 blocks of that one zone.”

124. Thus, in my opinion, Petitioner and Dr. Litz fail to show how Petitioner’s combination teaches “stor[ing] a value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host.”

5. Petitioner Does Not Show That Its Terasaki Combination Teaches A Memory Controller “provid[ing] to the host, responsive to a query received from the host, the information, as updated, and the stored value.”

125. Claim 1 recites, *inter alia*, a “storage device” comprising a “memory controller” having logic to cause the memory controller to “provide to the host, responsive to a ***query received from the host***, the information, as updated, and the stored value.” Petitioner does not show how this limitation is taught in its combination of Terasaki and Yamada.

126. Petitioner identifies Terasaki’s memory controller with the claimed “memory controller,” Pet., 67, and Terasaki’s host with the claimed “host.” *Id.*, 69.

For the claimed “information,” Petitioner identifies “information indicating the state of each zone (write state, erase state, erase request state, etc.)” and “information on the data stored in each zone (information on the physical block address of the block where the data is stored, information on the logical block address of the stored data, etc.).” *Id.*, 75 (quoting Ex. 1018 [Terasaki] [0049]). As noted above, *see* Section VI.B.4 *supra*, Petitioner does not expressly identify any “stored value” in Terasaki, but states:

When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet., 76 (bracketed insertion and boldface in original).

127. With these identifications, Petitioner does not explain how Terasaki’s controller “provide[s] to the host, responsive to a query received from the host, the information, as updated, and the stored value.”

128. Petitioner argues as follows:

Terasaki discloses that the controller and host exchange data, address information, status information [**provide to the host... the stored value**], and command information via host interface block 7 which includes a task file register storing addresses and commands, and an error register that may be set in the event of an error. Specifically,

status information is “fed into host system 4 using host interface block 7 as an outlet.” For instance, Terasaki discloses that to pass error information to the host, the host interface block that is used to pass information from the controller to the host may include an error register that may be set in the event of an error.

Pet., 79 (bracketed insertions and boldface in original) (citations omitted).

129. This argument fails to identify any alleged “query received from a host.” The relevant limitation involves a controller “provid[ing] to the host, responsive to a *query received from the host*, the information, as updated, and the stored value.” Petitioner’s statement that “the controller and host *exchange* data, address information, status information, and command information” does not identify any alleged “query received from the host” in Terasaki. A POSITA would recognize, for example, that Terasaki’s controller and Terasaki’s host can exchange information without any “query received from [its] host.”

130. This argument also fails to address Petitioner’s identification of previously recited “information.” “[T]he information” in the claimed “*provid[ing] to the host*, responsive to a query received from the host, *the information*” is a reference back to earlier-recited “information, for the specific one of the respective subdivisions, representing an extent to which the erase blocks of the specific one of the respective subdivisions are full.” Ex. 1001 [’801] cl. 1. Petitioner previously identified “information indicating the state of each zone (write state, erase state,

erase request state, etc.)” and “information on the data stored in each zone (information on the physical block address of the block where the data is stored, information on the logical block address of the stored data, etc.)” in Terasaki as the claimed “information.” Pet., 75. But here, Petitioner does not even argue, let alone show, that this earlier-identified “information” is provided to Terasaki’s host.

131. If Petitioner’s statement that “status information is ‘fed into host system 4 using host interface block 7 as an outlet,’” *id.*, 79 (quoting Ex. 1018 [Terasaki] [0024]), is an implicit reference back to Petitioner’s earlier-identified “information,” the cited support for Petitioner’s argument—Terasaki’s statement that “status information” is “fed into host system 4”—does not prove the point. This statement comes from a general disclosure in Terasaki that reads as follows:

Host interface block 7 is a functional block that exchanges data, address information, status information, and external command information with host system 4. In other words, when flash memory system 1 is mounted onto host system 4, flash memory system 1 and host system 4 are connected to each other via external bus 13. In such a state, data and the like that is supplied from host system 4 to flash memory system 1 is taken into the interior of controller 3 using host interface block 7 as an inlet, and data and the like that is supplied from flash memory system 1 to host system 4 is fed into host system 4 using host interface block 7 as an outlet.

Ex. 1018 [Terasaki] [0024]. Petitioner does not show how this disclosure relates to what it has previously identified as “information” being “provide[d] to the host, responsive to a query received from the host.”

132. Similarly, “the stored value” in the claimed “*provid[ing] to the host, responsive to a query received from the host ... the stored value*” is a reference back to earlier-recited “value representing a size of the NAND flash memory that is to be physically erased in fulfilling a maintenance request issued by the host.” As noted above, *see* Section VI.B.4 *supra*, Petitioner never identifies a “value” in Terasaki, but only states:

When data is rewritten, the written flag on the first page of the initial good block in the zone in which the original data is written is changed to an erase request flag for the up to 1000 blocks of that one zone [**store a value representing a size of the NAND flash memory that is to be physically erased**].

Pet., 76 (bracketed insertion and boldface in original).

133. Assuming for purposes of argument that this discussion of “erase request flag[s]” in Terasaki *did* somehow identify a “value,” Petitioner’s argument in connection with the present limitation makes no reference to any “erase request flags,” or to blocks in zones in which data was rewritten, or to otherwise relate what is allegedly “provide[d] to the host” to this “value.” *See id.*, 79.

134. Thus, in my opinion, Petitioner and Dr. Litz fail to show a device having logic to cause a memory controller to “provide to the host, responsive to a *query received from the host*, the information, as updated, and the stored value” in Petitioner’s combination of Terasaki and Yamada.

VII. CONCLUSION

135. In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required, I will appear for cross-examination within the United States during the time allotted.

136. I hereby declare that all statements made herein of my own knowledge are true and all statements made herein on information and belief were and are believed by me to be true, and that all statements herein were and are made with the 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 and that any such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Respectfully submitted,

Dated: November 5, 2025



Dr. Mark Jones

Exhibit A

**GENERAL
INFORMATION****Home Address:**

9121 NW 115th Terrace
Yukon, OK 73099
(540) 921-7477
marktjones25@me.com

US Citizen Born March 15, 1965 in Newport News, VA
Residence history: Newport News VA; Redlands CA; Dallas TX;
 Blacksburg VA; Clemson SC; Durham NC; Chicago IL;
 Knoxville TN; Blacksburg VA, Yukon OK

**PROFESSIONAL
EXPERIENCE**

Professor Emeritus of Electrical and Computer Engineering, Virginia Tech,
2021-present

Professor, Department of Electrical and Computer Engineering, Virginia Tech,
8/97-2021

Chief Technology Advisor, *Ocean Tomo, LLC*, August 2013 to December 2013.

Professor, Department of Computer Science, University of Tennessee, 8/93-7/97

Assistant Computer Scientist, Mathematics and Computer Science Division,
Argonne National Laboratory, Chicago, IL, 6/90 to 7/93.

EDUCATION

Ph.D., Duke University, Computer Science, May 1990.

B.S., Clemson University, Computer Science, summa cum laude, May 1986.

**RECOGNITION
AND AWARDS**

Virginia Bioinformatics Faculty Fellow
1st prize and Honorable Mention at AOL-CIT Research Day
Virginia Tech College of Engineering Dean's List for Teaching
NSF CAREER Award
University of Tennessee Science Alliance Award
Atanasoff Best Paper Award
Gordon Bell Prize
Von Neumann Fellowship
R. F. Poole Scholarship
National Merit Scholar

**EXPERT
TESTIMONY**

Blackboard v. Desire2Learn
United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving Internet-based educational
software employing role-based access control
Testified: Jury trial (infringement and validity), injunction hearing,
contempt hearing, and deposition (4 times)

TechRadium v. Blackboard
United States District Court for the Eastern District of Virginia

United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving network-based emergency notification systems
Testified: Preliminary injunction hearing and deposition

Computer Associates v. Digex
Circuit Court for Prince George's County Maryland
Overview: Contract dispute involving enterprise management systems
Testified: Jury trial and deposition

Computer Associates v. Retired Persons Services
United States District Court for the Eastern District of Virginia
Overview: Contract dispute involving enterprise management systems
Testified: Deposition

DVSI v. University of Phoenix and Walden University
United States District Court for the Eastern District of Virginia
Overview: Patent infringement case involving network-based educational software employing digital rights management
Testified: Deposition (3 days)

VirnetX v. Microsoft
United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving VPN software in Microsoft Windows and Microsoft's Unified Communications software
Testified: Jury trial (infringement and validity) and deposition (3 days)

TechRadium v. Twitter
United States District Court for the Southern District of Texas
Overview: Patent infringement case involving network-based notification systems
Testimony: Deposition

Bedrock Computer Tech v. Softlayer et al
United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving data structures in the Linux operating system's network stack
Testimony: Jury trial v. Google (infringement and validity), Jury trial v. Yahoo (infringement and validity) and deposition (3 days)

CEATS v. Continental Airlines et al
United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving Web-based interactive seat maps
Testimony: Jury trial (infringement and validity) and deposition (2 days)

VirnetX v. Cisco et al
United States District Court for the Eastern District of Texas
Overview: Patent infringement case involving secure communications, VPNs, and Voice-over-IP
Testimony: Jury trial v. Apple (infringement and validity), Jury trial v. Cisco (validity), and deposition (3 days), 2nd jury trial v. Apple (infringement), 3rd jury trial v. Apple (infringement), 4th jury trial v. Apple (infringement), 5th jury trial v. Apple

Summit6 v. Research In Motion et al
United States District Court for the Northern District of Texas

Overview: Patent infringement case involving network-based photo uploading tools on smartphones and web browsers

Testimony: Jury trial v. Samsung (infringement and validity) and deposition (3 days)

Ericsson v. Samsung

International Trade Commission

Overview: Patent infringement case involving Java technology and touchscreen technology on smartphones

Testimony: Deposition

AT Engine Controls LTD. v. Goodrich Corp and Goodrich Pump & Engine Control Systems, Inc.

United States District Court for the District of Connecticut

Overview: Trade secret case involving an embedded computing system design for helicopter engine control

Testimony: Deposition

SmartFlash LLC. v. Apple, et. al.

SmartFlash LLC. v. Samsung, et. al.

United States District Court for the Eastern District of Texas

Overview: Patent infringement case involving digital rights management and content distribution

Testimony: Jury trial v. Apple (infringement and validity) and deposition (4 days)

Unwired Planet v. Apple

United States District Court for the Northern District of California

Overview: Patent infringement case involving several technologies related to mobile devices

Testimony: Deposition (3 days); technology tutorial presentation to the Court

Facebook v. Rembrandt Social Media

Patent Trial and Appeal Board

Overview: Validity of a patent in the area of a web-based diary system

Testimony: Deposition

Motorola Mobility v. Intellectual Ventures

Patent Trial and Appeal Board

Overview: Validity of a patent in the area transmitting and browsing preformatted information

Testimony: Deposition

Summit6 v. Apple et al

United States District Court for the Northern District of Texas

Overview: Patent infringement case involving network-based photo uploading tools on smartphones and web browsers

Testimony: Deposition

EMC v. Pure Storage et al

United States District Court for the District of Delaware

Overview: Patent infringement case involving memory systems

Testimony: Deposition and jury trial (infringement and validity)

Parallel Networks v. IBM

United States District Court for the District of Delaware

Overview: Patent infringement case involving web servers and load balancing

Testimony: Deposition (2 days)

Parallel Networks v. Microsoft

United States District Court for the District of Delaware

Overview: Patent infringement case involving web servers and load balancing

Testimony: Deposition (2 days) and jury trial (infringement and validity)

Microsoft and IBM v. Parallel Networks

Patent Trial and Appeal Board

Overview: Validity of a patent related to web servers and load balancing

Testimony: Deposition

adidas v. Under Armour

United States District Court for the District of Delaware

Overview: Patent infringement case involving fitness tracking

Testimony: Deposition

BMC Software v. ServiceNow

United States District Court for the Eastern District of Texas

Overview: Patent infringement case involving enterprise management software

Testimony: Deposition

Ericsson v. TCL

United States District Court for the Eastern District of Texas

Overview: Patent infringement case involving Java technology and security on smartphones

Testimony: Deposition (3 days) and jury trial (infringement)

REM v. Square

United States District Court for the Eastern District of Missouri

Overview: Intellectual property case involving credit card readers for smartphones

Testimony: Deposition

AARC v. Chrysler et al

United States District Court for the District of Columbia

Overview: Intellectual property case involving car infotainment systems

Testimony: Deposition

Spex Technologies v. Apricorn

United States District Court for the Central District of California

Overview: Patent infringement case involving secure memory devices

Testimony: Deposition, Jury trial

United States Conference of Mayors v. Great-West Life Annuity Insurance Co.

United States District Court for the District of Columbia

Overview: Contract dispute

Testimony: Deposition and jury trial

Tecsec, Inc. v. Adobe Systems Inc. and Oracle (listed as IBM, et.al.).

United States District Court for the Eastern District of Virginia

Overview: Patent infringement case involving secure object technology

Testimony: Adobe: Deposition and jury trial (infringement and validity); Oracle (deposition)

Egenera, Inc. v. Cisco Systems, Inc.

United States District Court for the District of Massachusetts

Overview: Patent infringement case involving an architecture for a processing platform

Testimony: Deposition, bench trial, and jury trial

Unified Patents v. Sound View Innovations

Patent Trial and Appeal Board

Overview: Validity of a patent related to concurrency control

Testimony: Deposition

Hulu v. Sound View Innovations

Patent Trial and Appeal Board

Overview: Validity of a patent related to concurrency control

Testimony: Deposition

Grace and Potter. v. Apple Inc.

United States District Court for the Northern District of California

Overview: Class action case involving FaceTime

Testimony: Deposition

HTC v. Ericsson

United States District Court for the Eastern District of Texas

Overview: Patent licensing related to cellular standards

Testimony: Deposition

Windy City.Innovations v. FaceBook, Inc..

United States District Court for the Northern District of California

Overview: Patent infringement case involving an architecture for a messaging platform

Testimony: Deposition

Hulu v. Sound View Innovations

Patent Trial and Appeal Board

Overview: Validity of a patent related to real time systems

Testimony: Deposition

Citrix Systems, Inc. v. Avi Networks, Inc..

United States District Court for the District of Delaware

Overview: Patent infringement case involving network devices

Testimony: Deposition

Rovi v. Comcast

International Trade Commission

Overview: Patent infringement case set top boxes

Testimony: Deposition, Trial

Seven Networks v. Apple

United States District Court for the Eastern District of Texas

Overview: Patents related to mobile device push-based communication

Testimony: Deposition

Match Group, LLC v. Bumble Trading Inc.

United States District Court for the Western District of Texas

Overview: Patents related to online dating apps

Testimony: Deposition

SAS Institute Inc. v. World Programming Limited, et. al.
United States District Court for the Eastern District of Texas
Overview: Copyright and patents related to statistical software
Testimony: Deposition, Bench trial

Allscripts Healthcare, LLC v. Decision Resources Group,
United States District Court for the District of Massachusetts
Overview: Contract dispute
Testimony: Deposition, jury trial

Clear Imaging v. Samsung
United States District Court for the Eastern District of Texas
Overview: Patents related to image processing
Testimony: Deposition

Nokia v. Lenovo
International Trade Commission
Overview: Patents related to video encoding
Testimony: Deposition

Dish Network v. Sound View Innovations
Patent Trial and Appeal Board
Overview: Validity of a patent related to scheduling for QoS
Testimony: Deposition

Dish Network v. Sound View Innovations
Patent Trial and Appeal Board
Overview: Validity of a patent related real-time multimedia applications
Testimony: Deposition

RFCyber v. Samsung
United States District Court for the Eastern District of Texas
Overview: Patents related to secure payment systems
Testimony: Deposition

RightQuestion LLC v. Samsung
United States District Court for the Eastern District of Texas
Overview: Patents related to hardware-based authentication
Testimony: Deposition

Daedalus Blue, LLC v. MicroStrategy Inc.
United States District Court for the Eastern District of Virginia, Norfolk
Division
Overview: Patent infringement case involving business intelligence software
Testimony: Deposition

Seven Networks v. Motorola Mobility LLC,
United States District Court for the Eastern District of Texas
Overview: Patents related to mobile device communication methods
Testimony: Deposition

HID Global Corp. v. Vector Flow Inc. and Ajay Jain, Vikrant Ghai, and
Shailendra Sharma
United States District Court for the District of Delaware
Overview: Patents related to security monitoring

Testimony: Deposition and Trial

YouToo Technologies, LLC, v. Twitter, Inc.

United States District Court for the Northern District of Texas Dallas Division

Overview: Patents related to video

Testimony: Deposition (2 days) and jury trial

Constellation Designs, LLC v. LG Electronics, Inc., LG Electronics USA, Inc.,
and LG Electronics Alabama, Inc.

U.S. District Court, Eastern District of Texas, Marshall Division

Overview: Patents related to digital communication systems

Testimony: Deposition and Trial

Ryan Hardin, et al. v. Samsung Electronics Co., LTD., et al.

U.S. District Court, Eastern District of Texas, Marshall Division.

Overview: Patents related to geofencing

Testimony: Deposition

Xerox, ACS Transport Solutions, Conduent and New Jersey Transit v.

Bytemark, Inc.

Patent Trial and Appeal Board

Overview: Validity of a patent related to mobile ticketing

Testimony: Deposition

Motista LLC v. Ernst and Young LLP

Arbitration Proceedings

Overview: Matters related to trade secrets

Testimony: Hearing

Inovalon v. Komodo Health

Arbitration Proceedings

Overview: Matters related to trade secrets and breach of contract

Testimony: Hearing

Alacritech, Inc. v. Dell Inc. et al

U.S. District Court, Eastern District of Texas, Marshall Division

Overview: Patents related to network interface devices

Testimony: Deposition

RFCyber v. Apple

United States District Court for the Western District of Texas

Overview: Patents related to secure payment systems

Testimony: Deposition

K. Mizra LLC v. Hewlett Packard Enterprise Co and Aruba Networks LLC

U.S. District Court, Eastern District of Texas, Marshall Division

Overview: Patents related to network security

Testimony: Deposition

Carbyne v. Apple

United States District Court for the Western District of Texas

Overview: Patents related to hardware-based authentication

Testimony: Deposition

Radian Memory Systems LLC, v. Samsung Electronics Co., LTD., et al.

U.S. District Court, Eastern District of Texas, Marshall Division.

Overview: Patents related to flash memory controllers

Testimony: Deposition

Leo Riot, et al. v. Apex Trader Funding Inc., et al.

United States District Court for the Western District of Texas, Austin Division.

Overview: Matters related to trade secrets, copyrights, and breach of contract

Testimony: Deposition

PUBLICATIONS**Journals**

- R. Younes, M. Jones, T. Martin, "Classifier for Activities with Variations," *Sensors*, vol. 18(10), 19 pages, October 2018.
- M. Blake, R. Younes, D. Jacob, T. Martin, and M. Jones, "A User-Independent and Sensor -Tolerant Wearable Activity Classifier," *IEEE Computer*, v. 48, No. 10, October 2015, pp. 64-71.
- Hawkins JB, Jones MT, Plassmann PE, Thorley-Lawson DA, "Chemotaxis in Densely Populated Tissue Determines Germinal Center Anatomy and Cell Motility: A New Paradigm for the Development of Complex Tissues," *PLoS ONE* 6(12): e27650. doi:10.1371/journal.pone.0027650, 2011.
- Z. Nakad, M. Jones, T. Martin, and W. Fawaz, "Networking in E-textiles," *Computer Communications*, vol. 33, issue 6, April 2010, pp. 655-666.
- Jian Liu, T. E. Lockhart, M. Jones, and T. Martin, "Local Dynamic Stability Assessment of Motion Impaired Elderly Using Electronic Textile Pants," *IEEE Transactions on Automation Science and Engineering*, vol. 5, issue 4, pp. 696-702, October 2008.
- Madhup Chandra, Mark Jones, and Thomas Martin, "E-Textiles for Autonomous Location Awareness," *IEEE Transactions on Mobile Computing*, vol. 6, issue 4, pp. 367-380, April 2007.
- Zahi Nakad, Mark Jones, Thomas Martin, and Ravi Shenoy, "Using Electronic Textiles to Implement an Acoustic Beamforming Array: A Case Study," *Pervasive and Mobile Computing Journal*, vol. 3, issue 5, pp. 581-606, October 2007.
- Mark Jones, Zahi Nakad, Paul Plassmann, Yanhua Yi, "The Use of Configurable Computing for Computational Kernels in Scientific Simulations," *Intern. Journal of Future Generation Computer Systems*, 22 (1-2), pp. 67-79 (2006).
- J.-R. Cheng, M. T. Jones, and P. E. Plassmann, "A Portable Software Architecture for Mesh-Independent Particle Tracking Algorithms," *Journal of Parallel Algorithms and Applications*, 19 (2-3), 145-161, 2004.
- Jae H. Park, Gary Friedman and Mark Jones, "Geographical Feature Sensitive Sensor Placement," *Journal of Parallel and Distributed Computing*, volume 64, 2004, pp. 815-825.
- D. Marculescu, R. Marculescu, N. Zamora, P. Stanley-Marbell, P. K. Khosla, S. Park, S. Jayaraman, S. Jung, C. Lauterbach, W. Weber, T. Kirstein, D. Cottet, J. Grzyb, G. Tröster, M. Jones, T. Martin, Z. Nakad, "Electronic Textiles: A Platform for Pervasive Computing," *Proceedings of the IEEE*, volume 91, number 12, December 2003, pp. 1995-2018.
- Kiran Puttegowda, David I. Lehn, Jae H. Park, P. Athanas and Mark Jones, "Context Switching in a Run-Time Reconfigurable System," *Journal of Supercomputing*, Kluwer Academic Press, June 2003, pp 239-257.
- Mark Jones, Shashank Mehrotra, and Jae Hong Park, "Tasking Distributed Sensor Networks," *Journal of High Performance Computing Applications*, Vol 16, pp. 243-257, 2002.
- Eloise Coupey and Mark Jones, "A Script-Based Approach for E-Commerce Applications," *Quarterly Journal of Electronic Commerce*, to appear.

Eloise Coupey and Mark Jones, "Decision Making in the Electronic Commerce Environment: Issues and Approaches for Tool Development," *Quarterly Journal of Electronic Commerce*, Vol 1, pp. 215-228, 2000.

Mark Jones and Karthik Ramachandran, "Unstructured Mesh Computations on CCMs," *Advances in Engineering Software*, Vol. 31, pp. 571-580, 2000.

Mark Jones and Paul Plassmann, "Unstructured Mesh Computations on Networks of Workstations," *Computer-Aided Civil and Infrastructure Engineering*, Vol. 15, 196-208, 2000.

Lori Freitag, Mark Jones, and Paul Plassmann, "A Parallel Algorithm for Mesh Smoothing," *SIAM Journal on Scientific Computing*, Vol 20, pp 2023-2040, 1999.

William Barry, Mark Jones, and Paul Plassmann, "Parallel Adaptive Mesh Refinement Techniques for Plasticity Problems," *Advances in Engineering Software*, Vol. 19, pp. 217-229, 1998.

Mark Jones and Paul Plassmann, "Adaptive Refinement of Unstructured Finite-Element Meshes," *Journal of Finite Elements in Analysis and Design*, Vol. 25, pp. 41-60, 1997.

Mark Jones and Paul Plassmann, "Parallel Algorithms for Adaptive Mesh Refinement," *SIAM Journal of Scientific Computing*, Vol. 18, pp. 686-708, 1997.

Robert Gjertsen, Mark Jones, and Paul Plassmann, "Parallel Heuristics for Improved, Balanced Graph Colorings," *Journal of Parallel and Distributed Computing*, Vol. 37, pp. 171-186, 1996.

Mark Jones and Daniel Szyld, "Two-stage Multisplitting Methods with Overlapping Blocks," *Numerical Linear Algebra with Applications*, Vol. 3, pp. 113-124, 1996.

Mark Jones and Paul Plassmann, "An Improved Incomplete Cholesky Factorization," *ACM Trans. on Mathematical Software*, Vol. 21, pp. 5-17, 1995.

Mark Jones and Paul Plassmann, "Algorithm 740: Fortran Subroutines to Compute Improved Incomplete Cholesky Factorizations," *ACM Trans. on Mathematical Software*, Vol. 21, pp. 18-19, 1995.

Mark Jones and Paul Plassmann, "Results for Parallel Unstructured Mesh Computations," *Computing Systems in Engineering*, Vol. 5, pp. 297-309, 1994.

Mark Jones and Paul Plassmann, "Scalable Iterative Solution of Sparse Linear Systems," *Parallel Computing*, Vol. 20, pp. 753-773, 1994.

Mark Jones and Merrell Patrick, "Factoring Indefinite Matrices on High-Performance Architectures," *SIAM Journal on Matrix Analysis and Applications*, Vol. 15, pp. 273-283, 1994.

Mark Jones and Paul Plassmann, "Computation of Equilibrium Vortex Structures for Type-II Superconductors," *Int. J. Supercomputing Applications*, Vol.7.2, pp. 129-143, 1993.

Mark Jones and Paul Plassmann, "A Parallel Graph Coloring Heuristic," *SIAM J. on Scientific and Statistical Computing*, Vol. 14, pp. 654-669, 1993.

Mark Jones and Merrell Patrick, "Bunch-Kaufman Factorization for Real Symmetric Indefinite Banded Matrices," *SIAM Journal of Matrix Analysis and Applications*, Vol. 14, pp. 553-559, 1993.

Mark Jones and Merrell Patrick, "The Lanczos Algorithm for the Generalized Symmetric Eigenproblem on Shared-Memory Architectures," *Applied Numerical Mathematics*, Vol. 12, pp. 377-389, 1993.

Daniel Szyld and Mark Jones, "Two-stage and Multi-splitting Methods for the Parallel Solution of Linear Systems," *SIAM Journal of Matrix Analysis and Applications*, Vol.13, pp. 671-679, 1992.

Conferences (refereed)

Alireza Golgouneh, Eric Beaudette, Heidi Woelfle, Bai Li, Niharikha Subash, Amanda Redhouse, Mark Jones, Tom Martin, Michele A. Lobo, Brad Holschuh, Lucy Dunne, "Design of a Hybrid SMA-Pneumatic based Wearable Upper Limb Exoskeleton," ISWC'21, September 21-26, 2021.

R. Younes, M. Jones and T. L. Martin, "Toward Practical Activity Recognition: Recognizing Complex Activities with Wide Variations," 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, 2018, pp. 9-14.

R. Younes, T. Martin, and M. Jones, "Activity Classification at a Higher Level: What to Do After the Classifier Does Its Best?" Proceedings of the International Symposium on Wearable Computers, Osaka, Japan, September 2015, pp. 83-86.

Joo Hong Lee, Mark T. Jones, and Paul E. Plassmann, "Scalable Solution of Radiative Heat Transfer Problems by the Photon Monte Carlo Algorithm on Hybrid Computing Architectures," Proceedings of the 2012 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'12: July 16-19, 2012, USA), Volume 1, pp. 67-75.

Joo Hong Lee, Mark T. Jones, Paul E. Plassmann, "A Hybrid Software Framework for the GPU Acceleration of Multi-Threaded Monte Carlo Applications," Proceedings of the 2010 International Conference on Scientific Computing (CSC'11: July 18-21, 2011, USA), pp. 70-76.

Joo Hong Lee, Mark T. Jones, Paul E. Plassmann, "An Efficient Shared Memory Programming Model for Biological Systems Simulation", 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, July 12-15, 2010, pp. 315-319.

Joo Hong Lee, Mark T. Jones, Paul E. Plassmann, "A Scalable Distributed Memory Programming Model for Large-Scale Biological Systems Simulation", 2010 International Conference on Scientific Computing, July 12-15, 2010, pp. 251-256.

Joo Hong Lee, Mark T. Jones, Paul E. Plassmann, "A Hybrid Parallel Programming Model for Large-Scale Biological Systems Simulation", Proceedings of Para 2010, State of the Art in Scientific and Parallel Computing, June 6-9, 2010.

M. Quirk, T. Martin, and M. Jones, "Inclusion of Fabric Properties in the E-Textile Design Process," *Proc. 13th Int'l Symp. Wearable Computers* (ISWC 09), IEEE CS Press, 2009, pp. 34-40.

Tom Martin, Mark Jones, Justin Chong, Meghan Quirk, Kara Baumann, Leah Passauer, "Design and Implementation of an Electronic Textile Jumpsuit," *Proc. 13th Int'l Symp. Wearable Computers* (ISWC 09), IEEE CS Press, 2009, pp. 157-158.

M. Shelburne, C. Patterson, P. Athanas, M. Jones, B. Martin, and R. Fong, "Metawire: Using FPGA Configuration Circuitry to Emulate a Network-on-

Chip”, *Field Programmable Logic and Applications*, September 2008, pp. 257-262.

M. Jones, T. Martin, and B. Sawyer, “An Architecture for Electronic Textiles”, *Proceedings of the ICST Third International Conference on Body Area Networks*, 2008, Article 24, 4 pages.

David Graumann, Giuseppe Raffa, Meghan Quirk, Braden Sawyer, Justin Chong, Mark Jones, Thomas Martin, “Large Surface Area Electronic Textiles for Ubiquitous Computing: A Systems Approach”, *MobiQuitous '07*, August 2007, pp. 1-8.

George Eichinger, Tom Martin, and Mark Jones, “From Circuit to Sewing in One Click,” *ISWC 2007*.

J. Edmison, D. Lehn, M. Jones, and T. Martin, An E-Textile Based Automatic Activity Diary for Medical Annotation and Analysis, *2006 Workshop on Body Sensor Networks*, April 2006, pp. 131-134.

C. Einsmann, M. Quirk, B. Muzal, B. Venkatramani, T. Martin, and M. Jones, “Modeling a Wearable Full-body Motion Capture System,” *Proceedings of the 2005 IEEE International Symposium on Wearable Computers (ISWC)*, October 2005, pp. 144-151.

J. Edmison, D. Lehn, M. Jones, and T. Martin, “Users’ Perceptions of an Automatic Activity Diary for Medical Annotation and Analysis,” *Proceedings of the 2005 IEEE International Symposium on Wearable Computers (ISWC)*, October 2005, pp. 192-193.

M. Chandra, M. Jones, and T. Martin, “E-Textiles for Autonomous Location Awareness,” *Proceedings of the 2004 International Symposium on Wearable Computers*, Arlington, VA, Oct. 31-Nov. 3, 2004, pp. 48-55.

Zahi Nakad, Mark Jones, and Tom Martin, “Fault Tolerant Networks for Electronic Textiles,” *CIC 2004*, Las Vegas, June 2004, pp. 100-106.

Zafer Gurdal, Tom Hartka, Mark Jones, and Sun Wook Kim, “A Reconfigurable Approach to Structural Engineering Design Computations,” *ERSA 2004*, Las Vegas, June 2004.

Jones, Mark T., and Eloise Coupey, "An Agent-based Simulation Prototype for Evaluating Health Behavior Interventions," *METMBS 2004*, Las Vegas, June 2004.

Thomas Martin, Mark Jones, Joshua Edmison, Tanwir Sheikh, and Zahi Nakad, “Modeling and Simulating E-Textile Applications *Proceedings of the ACM Conference on Languages, Compilers, and Tools for Embedded Systems*, June 11-13, 2004, pp. 10-19.

Lehn, D., C. Neely, K. Schoonover, T. Martin, and M. Jones, "e-TAGS: e-Textile Attached Gadgets." *Communication Networks and Distributed Systems Modeling and Simulation Conference*, January 2004.

M. Abdalla, S. W. Kim, Z. Gürdal, and M. T. Jones, "Multigrid Accelerated Cellular Automata for Design Optimization of Continuum Structures: A 1-D Implementation", *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, April 2004.

Zahi Nakad, Mark Jones, and Tom Martin, “Communication in Electronic Textile Systems,” *2003 International Conference on Communications in Computing (CIC 2003)*, pp. 37-43.

Mark Jones, Paul Plassmann, Zahi Nakad, and Yanhua Yi, "The Use of Configurable Computing in Scientific Simulations," *2003 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA '03)*, pp. 520-524.

Mark Jones, Mukta Nandwani, Jae Park, Paul Plassmann, and Yanhua Yi, "Examining the Communication Requirements of Remote Scientific Visualization," *2003 International Conference on Communications in Computing (CIC 2003)*, pp. 31-36.

T. Martin, M. Jones, J. Edmison, and R. Shenoy, "Towards a Design Framework for Wearable Electronic Textiles," *International Society for Wearable Computers 2003*, pp. 190-199.

M. Jones., T. Martin, Z. Nakad, R. Shenoy, T. Sheikh, D. Lehn, and J. Edmison, "Analyzing the Use of E-textiles to Improve Application Performance," *IEEE Vehicular Technology Conference 2003, Symposium on Wireless Ad hoc, Sensor, and Wearable Networks*.

Mark Jones, Tom Martin, and Zahi Nakad, "A Service Backplane for e-Textiles," *MAMSET 2002: Proc. Workshop on Modeling, Analysis and Middleware Support for Electronic Textiles*, 6 October 2002, pp. 15-22.

J. Edmison, M. Jones, Z. Nakad and T. Martin "Using piezoelectric materials for wearable electronic textiles," *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on* pp. 41-48, 2002.

Mark Jones, Lucas Scharf, Jon Scott, Christian Twaddle, Matthew Yaconis, Kuan Yao, Peter Athanas, and Brian Schott, "Implementing an API for Distributed Adaptive Computing Systems," *Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, CA, April, 1999).

Jason Hess, David Lee, Scott Harper, Peter Athanas, and Mark Jones, "Implementation of a Prototype Reconfigurable Router," *Proceedings of IEEE Symposium on Field-Programmable Custom Computing Machines*, Napa, CA, April, 1999, to appear 7 pages).

David Lee, Mark Jones, Scott Midkiff, and Peter Athanas, "Towards Active Hardware," *Lecture Notes in Computer Science 1653*, Springer-Verlag, pp. 180-187, 1999.

Lori Freitag, Mark Jones and Paul Plassmann, "The Scalability of Mesh Improvement Algorithms," *The IMA Volumes in Mathematics and its Applications*, Vol. 105, pp. 185-211, 1998.

Lori Freitag, Mark Jones, and Paul Plassmann, "An Efficient Parallel Algorithm for Mesh Smoothing," *Proceedings of the 4th International Meshing Roundtable*, Albuquerque, NM, pp. 47-58, 1995.

Mark Jones and Paul Plassmann, "The efficient parallel iterative solution of large sparse linear systems," *The IMA Volumes in Mathematics and its Applications*, Vol. 56, pp. 229-245, 1993.

Mark Jones and Paul Plassmann, "Solution of Large, Sparse Systems of Linear Equations in Massively Parallel Applications," *Supercomputing '92 Proceedings*, IEEE Computer Society, pp. 551-560, 1992.

Mark Jones, Merrell Patrick, and Robert Voigt, "A Language Comparison for Scientific Computing on MIMD Architectures," *Proceedings of the IFIP Working Conference: Aspects of Computation on Asynchronous Parallel Processors*, M. H. Wright (Editor), Elsevier Science Publishers B. V. (North-Holland), IFIP, pp. 55-67, 1989.

Book Chapters

R. Younes, K. Hines, J. Forsyth, J. Dennis, T. Martin, and M. Jones, "The design of smart garments for motion capture and activity classification," Chapter 27 in *Smart Textiles and Their Applications*, V. Koncar (editor), Woodhead Publishing, Duxford, UK, 2016, pp. 627-655.

M.T. Jones and T.L. Martin, "Hardware and Software Architectures for Electronic Textiles," a chapter in "Smart Clothing: Technology and Applications", ed. Gilsoo Cho, 2010, pp. 135-152.

Other

J. Edmison, M. Jones, T. Lockhart, and T. Martin, "An E-Textile System for Motion Analysis," *Wearable eHealth Systems for Personalised Health Management: State of the Art and Future Challenges*, Studies in Health Technology and Informatics, vol. 108, August 2004, pp. 292-301

Eloise Coupey and Mark Jones, Developing Dynamic Decision Support: Opportunities, Issues and Approaches, Twenty-Third Annual International Computer Software and Applications Conference, October 1999.

Mark Jones, Michael Langston, and Padma Raghavan, "Tools for Mapping Applications to CCMs," *Proceedings of SPIE: Configurable Computing: Technology and Applications*, Boston, MA, SPIE, pp. 72-80, Nov. 1998.

Lori Freitag, Mark Jones, and Paul Plassmann, "Mesh Component Design and Implementation within SUMAA3d," *Proceedings of SIAM Workshop on Object Oriented Methods for Interoperable Scientific and Engineering Computing*, SIAM Publications, pp. 215-224, 1998.

Lori Freitag, Mark Jones, and Paul Plassmann, "A Parallel Algorithm for Mesh Smoothing," *Proceedings of the Eighth SIAM Conference on Parallel Processing for Scientific Computing*, Minneapolis, MN, March 1997.

Lori Freitag, Mark Jones, and Paul Plassmann, "Parallel Adaptive Mesh Refinement with the SUMAA3d Project," *Proceedings of the ICASE/LaRC Workshop on Adaptive Grid Methods*, 1995.

Lori Freitag, Mark Jones, and Paul Plassmann, "Parallel Algorithms for Unstructured Mesh Computation," *Proceedings of Fifth SIAM Applied Linear Algebra Conference*, SIAM Publications, pp. 123-127, 1994.

Mark Jones and Paul Plassmann, "Parallel Algorithms for the Adaptive Refinement and Partitioning of Unstructured Meshes," *Proceedings of the Scalable High-Performance Computing Conference*, IEEE, Ed. Dongarra and Walker, pp. 478-485, 1994.

Lori Freitag, Mark Jones, and Paul Plassmann, "New Techniques for Parallel Simulation of High-Temperature Superconductors," *Proceedings of the Scalable High-Performance Computing Conference*, IEEE, Ed. Dongarra and Walker, pp. 726-733, 1994.

Lori Freitag, Mark Jones, and Paul Plassmann, "New Advances in the Modeling of High-Temperature Superconductors," *Proceedings of the 1994 International Simulation Conference – Grand Challenges in Computer Simulation*, The Society for Computer Simulation, pp. 208-213, 1994.

Mark Jones and Paul Plassmann, "Software for the Generalized Eigenproblem on Distributed Memory Architectures," *Proceedings of the Lanczos Centenary Conference*, Ed. Chu, et. al., pp. 322-325, 1994.

Lori Freitag, J. Garner, Mark Jones, Paul Plassmann, "Recent Computational Results on the Equilibrium Vortex Configurations on Type-II Superconductors," *Proceedings of the Second DELTA Applications Workshop*, Ed. P. Messina, pp. 93-98, March 1993.

Tom Canfield, Mark Jones, Paul Plassmann, and Michael Tang, "Modeling Piezoelectric Crystals on the Intel DELTA," *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pp. 156-159, 1993.

Mark Jones and Paul Plassmann, "Recent Results in the Modeling of Type-II Superconductors on Massively Parallel Computers," *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pp. 147-151, 1993.

Mark Jones and Paul Plassmann, "Parallel Solution of Unstructured, Sparse Systems of Linear Equations," *Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing*, pp. 471-475, 1993.

Tom Canfield, Mark Jones, Paul Plassmann, and Michael Tang, "Thermal Effects on the Frequency Response of Piezoelectric Crystals," *New Methods in Transient Analysis*, Eds. P. Smolinski, W. K. Liu, G. Hulbert, and K. Tamma, PVP-Vol. 246 and AMD-Vol. 143, pp. 103-108, ASME, New York, 1992.

Mark Jones and Paul Plassmann, "The Effect of Many Color Orderings on the Convergence of Iterative Methods," *Proceedings of the Copper Mountain Conference on Iterative Methods*, April 1992.

Mark Jones and Paul Plassmann, "Modeling piezoelectric crystals on the Intel DELTA," *Proceedings of the Intel DELTA Applications Workshop*, Cal Tech, pp. 97-105, 1992.

RESEARCH GRANTS NASA, "Phase II: Integrated Sensors for the Evaluation of Structural Integrity of Inflatable Habitats," with NanoSonic, \$180,000, 9/25/17-9/24/19.

NSF, "SCH: INT: Collaborative Research: Smart Wearable Systems to Support and Measure Movement in Children With and Without Mobility Impairments," \$326,658, 10/1/17-9/30/21.

NASA, "Integrated Sensors for the Evaluation of Structural Integrity of Inflatable Habitats," with NanoSonic, \$35,000, 9/1/2016-8/31/17.

NSF, "SHB: Small: Collaborative Research: Electronic textiles for ambulatory health monitoring," \$306,083, 7/1/2011-6/30/2014.

DARPA, "RECE," With BBN, \$30,000, 4/16/09-10/16/09.

NSF, "Investigating a Novel Embedded Processor Architecture for Electronic Textiles in Wearable and Pervasive Computing," \$220,000, 12/1/08-11/30/10.

NSF, "Fundamental Algorithms to Enable the Simulation of Multi-Scale Biological Systems," \$200,000, 9/1/07-8/31/10.

AFRL, "Phase II: Amorphous Soft-Core Processor for Hardware Anti-Tamper," ~\$230,000, 11/1/06-10/31/08.

Intel, "E-Textile Rug for Gait Analysis, People Tracking and Emergency Directions," \$15,000, 6/1/06-12/31/06.

Carilion, "Quantitative Measurement and Modeling of Early Events in Influenza Pathogenesis," \$20,000, 7/1/05-6/30/06.

AFRL, "Phase I: Wearable Computer for Enhanced Situation Awareness," \$30,000, 5/1/06-1/1/07.

AFRL, "Phase I: Amorphous Soft-Core Processor for Hardware Anti-Tamper," \$35,000, 1/1/06-11/1/06.

MDA, "Phase I: Secure Software Platform for Real-Time Software Anti-Tamper," \$29,994, 6/1/06-11/1/06.

National Science Foundation (NSF), "CRI", (e-textiles equipment), ~\$80,000, 8/05-7/07.

Harris Corporation, "Exploiting the Reconfigurability of a Software-Defined Radio Platform," \$161,070, 9/1/05-6/30/06.

DARPA, "Phase I: Technology for Trusted Circuits," \$31,000, 3/1/05-9/15/05.

Harris Corporation, "Partial Reconfiguration Support for the Harris Programmable Modem Platform," \$73,448, 12/01/04-6/30/05.

NSF, "Phase I: An Electronic Textile System for Gait Analysis", \$33,000, 1/1/2005-6/30/2005.

Office of Naval Research (ONR). "AWINN." \$452,300, 12/20/04-7/31/06.

Office of Secretary of Defense, "Phase II: Reconfigurable Processor Technology for Software Protection," SBIR Subcontract with Luna Innovations, \$237,581, 6/04-5/06.

NSF, REU Supplement to ITR: Tailor-Made: Design of e-Textile Architectures for Wearable Computing, NSF, \$12,000, 6/1/2005 – 8/31/2005.

Office of Secretary of Defense, "Phase I: Reconfigurable Processor Technology for Software Protection," SBIR Subcontract with Luna Innovations, \$33,000, 8/03-2/04.

National Science Foundation (NSF), "ITR: Tailor-Made: Design of e-Textile Architectures for Wearable Computing", \$399,000, 8/02-8/06.

NSF, REU Supplement to above grant, \$10,000.

NSF, REU Supplement to above grant, \$6,000.

NSF, "A Toolbox of Scalable Algorithms and Software for Advanced Scientific Computing Applications", \$300,000, 4/03-4/06.

Defense Advanced Research Projects Agency (DARPA). Computational Fabrics, \$217,000, 3/01-12/02. Subcontract from ISI/University of Southern California.

Office of Naval Research (ONR). Secure Configurable Radio. \$751,000, 4/00-4/04.

NSF, SUCCEED, \$33,000, 9/1/01-8/31/02.

National Science Foundation (NSF). Parallel unstructured mesh algorithms, Amount: \$145,023, 10/99-9/02.

DARPA, Development of algorithms and software for the control of distributed, dynamic sensor networks, Amount \$411,625, 6/99-6/02.

DARPA, Subcontract with USC/ISI. Development of algorithms and runtime software for systems of configurable computers, Amount \$160,000, 10/00-10/01.

Department of Energy, Development of algorithms and software for massively parallel architectures, Amount: \$145,000, 12/98-11/01.

DARPA, Development of Java-based tools and applications for fast, flexible run-time reconfiguration of configurable computing devices, Amount: \$345,574, 9/99-9/01.

DARPA, Development of an application programming interface, algorithms, and applications for run-time reconfiguration on novel adaptive computing architectures, Amount: \$250,000, 6/99-6/01.

Lockheed-Martin/Sanders, A Wireless Link for Remote Telemetry of Compact Airborne System, \$125,000, 5/00-4/01.

DARPA, Development of algorithms and runtime software for systems of configurable computers, Amount \$169,792, 11/99-10/00.

National Security Agency (NSA), Software for the translation of JBits FPGA programs to EDIF programs, Amount \$75,000, 9/99-7/00.

DARPA, Construction of an API & applications for the control of distributed systems of configurable computing nodes, Amount \$162,886, 9/98-10/99.

NSF, Career Award: Parallel algorithms and software for unstructured mesh computations, Amount: \$125,881, 7/96-7/99.

Air Force Research Laboratory, Research and development of an Internet-based, interactive, decision making program, Amount: \$97,425, 11/97-5/99.

NSF, Equipment funding for a 16-node cluster of configurable computers, Amount: \$85,000, 12/97-12/98.

NSF, Scientific applications in a distributed computing environment, Amount: \$318,807, 7/95-7/98.

NSF, Equipment funding for an ATM-based cluster of workstations, Amount, \$100,000, 1/96-12/96.

NSF, Workshop organized at Argonne National Laboratory, Amount, \$10,000, 1996.

PATENTS

Inventor, United States Patent No. 8,473,754, Hardware-Facilitated Secure Software Execution Environment.

SOFTWARE

SLAAC/ACS API (4/99): Software for the control of complex systems of configurable computing nodes and high-speed networks. This code has been distributed to the ACS community. Developed in Configurable Computing Laboratory with Peter Athanas.

BlockSolve (4/93 & 1/96): Software for solving large sparse linear systems on distributed memory architectures. This code has been distributed via netlib and ANL ftp and has been accessed by several hundred people as of 4/4/99. Developed at Argonne National Laboratory with Paul Plassmann.

LANZ (10/90): Software for solving the generalized eigenproblem on shared memory architectures. This code has been distributed via netlib and NASA and has been accessed by over 10,000 people as of 4/4/99. Developed at Duke University and Argonne National Laboratory with Merrell Patrick.

TEACHING

Computer Programming

RESPONSIBILITIES

ECE 5574, *Advanced Software Development for CAE*
ECE 4574, *Large-Scale Software Development for Engineering Systems*
ECE 3574, *Applied Software Design*
ECE 2574, *Data Structures*
ECE 1574, *Engineering Problem Solving with C++*
Computer Simulation and Modeling
CS 371, *Numerical Analysis*
CS 594, *Computational Modeling*
Computer Network, Architecture, and Organization
ECE 2504, *Introduction to Computer Engineering*
CS 594, *Internetworking with TCP/IP*
ECE 4504, *Computer Organization*
ECE 4534, *Design of Embedded Systems*
ECE 5504 (formerly ECE 5515), *Computer and Network Architectures*
ECE Special Studies: *Configurable Computing*, E-Textiles
ECE 5984, *Java-Based Configurable Computing*
ECE 6504, *Applications of Parallel and Distributed Computing*, TV course.
CS 530, *Computer Systems Organization*
CEO of DISC, a student-run virtual corporation, for Fall 1998 and Spring 1999.
