

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

BMW OF NORTH AMERICA, LLC,

Petitioner,

v.

FORAS TECHNOLOGIES LTD.,

Patent Owner.

Case IPR2024-01347
U.S. Patent 7,624,302

PATENT OWNER’S RESPONSE TO THE PETITION

Mail Stop “PATENT BOARD”
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	Introduction.....	1
II.	Background	1
III.	Burden of Proof.....	5
IV.	Person of Ordinary Skill in the Art	7
V.	Statement of Facts	8
VI.	The Challenged Claims of the '302 Patent are Not Unpatentable.	11
A.	Grounds 1 and 2 do not establish that challenged dependent claims are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.....	13
B.	Ground 1 does not establish that challenged independent claims 1, 13 and 21 are rendered obvious by Fox in view of Safford and Arai.....	15
1.	The Prior Art in Ground 1.....	15
2.	Petitioner has not met its burden to establish claim 1 is unpatentable.....	20
3.	Petitioner has not met its burden to establish claim 1's limitation of switching the role of boot processor to a spare processor would have been obvious.	24
C.	Ground 1 does not establish that challenged independent claims 13 and 21 are rendered obvious by Fox in view of Safford and Arai.....	30
D.	Ground 2 does not establish that challenged independent claims 1, 13 and 21 are rendered obvious by Fox in view of Safford, Landry and Arai.....	31

1. The Additional Prior Art in Ground 2.....	31
2. Petitioner has not met its burden to establish claim 1 is unpatentable in Ground 2.	33
3. Petitioner has not met its burden to establish claims 13 and 21 are unpatentable in Ground 2.....	35
E. Grounds 3 and 4 do not establish that challenged independent claim 26 or dependent claim 27 are rendered obvious by Fox in view of Safford, Arai and Bigbee or Fox in view of Safford, Landry, Arai, and Bigbee.....	37
F. Grounds 1 and 2 do not establish that challenged dependent claims 3, 4, 15, 16, and 24 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.	38
G. Grounds 1 and 2 do not establish that challenged dependent claims 6, 8, and 25 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.	45
H. Grounds 1 and 2 do not establish that challenged dependent claims 11, 12, 17 and 24 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.....	48
VII. Conclusion	50

TABLE OF AUTHORITIES

Cases

<i>CA, Inc. v. Simple.com, Inc.</i> , 780 F. Supp. 2d 196 (E.D.N.Y. 2009)	14
<i>Comaper Corp. v. Antec, Inc.</i> , 596 F.3d 1343 (Fed. Cir. 2010)	13
<i>Duhn Oil Tool, Inc. v. Cooper Cameron Corp.</i> , 2012 U.S. Dist. LEXIS 24088, 2012 WL 604138 (E.D. Cal. Feb. 23, 2012)	14
<i>Harmonic Inc. v. Avid Tech., Inc.</i> , 815 F.3d 1356 (Fed. Cir. 2016)	6
<i>In re Fine</i> , 837 F.2d 1071 (Fed. Cir. 1988)	13
<i>In re Magnum Oil Tools Int'l, Ltd.</i> , 829 F.3d 1364 (Fed. Cir. 2016)	48
<i>In re NuVasive Inc.</i> , 841 F.3d 966 (Fed. Cir. 2016)	48
<i>Intelligent Bio-Systems, Inc. v. Illumina Cambridge Ltd.</i> , 821 F.3d 1359 (Fed. Cir. 2016)	6, 49
<i>Microsoft Corp. v. Biscotti, Inc.</i> , 878 F.3d 1052 (Fed. Cir. 2017)	50
<i>Netflix, Inc. v. DivX, LLC</i> , 84 F.4th 1371 (Fed. Cir. 2023)	48, 49
<i>Oren Techs., LLC v. Proppant Express Invs. LLC</i> , 2021 U.S. App. LEXIS 21859, 2021 WL 3120819 (Fed. Cir. July 23, 2021)	49
<i>Stryker Corp. v. Karl Storz Endoscopy America, Inc.</i> , IPR2015-00677, Paper 15 (PTAB Sept. 2, 2015)	13, 24

<i>SynQor, Inc. v. Artesyn Techs., Inc.</i> , 709 F.3d 1365 (Fed. Cir. 2013).....	13
<i>Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.</i> , 853 F.3d 1272 (Fed. Cir. 2017).....	50
<i>Zimmer Surgical, Inc. v Stryker Corp.</i> , 365 F.Supp. 3d 466 (D. Del. 2019).....	13

Statutes

17 U.S.C. § 401.....	23
35 U.S.C. § 312.....	6, 49
35 U.S.C. § 316.....	5

Rules

37 C.F.R § 42.104	7, 49
-------------------------	-------

TABLE OF PATENT OWNER'S EXHIBITS

Exhibit No.	Description
2001	U.S. Patent No. 7,251,746 ("Fox"), dated July 31, 2007
2002	U.S. Patent No. 7,085,959 ("Safford"), dated August 1, 2006
2003	U.S. Patent No. 6,438,687 ("Klein"), dated August 20, 2002
2004	Declaration of Nigel A. Jones
2005	Curriculum Vitae of Nigel A. Jones
2006	Published Patent Application US 2006/0107112 A1
2007	Deposition Transcript of Dr. Michael Brogioli

LISTING OF CLAIMS OF U.S. PATENT NO. 7,624,302

Claim 1	
[1pre]	A method, comprising:
[1a]	detecting loss of lockstep (LOL) for a processor in a multi-processor system;
[1b]	determining that the processor for which said LOL is detected is assigned a role of boot processor;
[1c]	changing a system identifier, used by the system's operating system to recognize the boot processor, of said boot processor for which said LOL is detected, to a different value, which causes the system's operating system to not recognize the processor for which said LOL is detected as the boot processor; and
[1d]	changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to a spare processor without shutting down the system's operating system.
Claim 2	
2	The method of claim 1 wherein the spare processor is a hot spare that is held in idle until switched the role of boot processor.
Claim 3	
3	The method of claim 1 wherein the spare processor is an active spare.
Claim 4	
4	The method of claim 3 wherein said switching further comprises: causing the system's operating system to idle the active spare.
Claim 5	
5	The method of claim 1 wherein the spare processor is a processor module that includes a master and slave processors that operate in lockstep.
Claim 6	
6	The method of claim 1 wherein said determining that the processor for

	which said LOL is detected is assigned the role of boot processor comprises: accessing information stored to non-volatile data storage that identifies one of the processors in said multi-processor system that is assigned the role of boot processor.
Claim 7	
7	The method of claim 1 further comprising: determining the processor that is assigned the role of spare processor.
Claim 8	
8	The method of claim 7 wherein said determining the processor that is assigned the role of spare processor comprises: accessing information stored to non-volatile data storage that identifies one of the processors in said multi-processor system that is assigned the role of spare processor.
Claim 9	
9	The method of claim 1 wherein said determining step is performed by system firmware.
Claim 10	
10	The method of claim 1 wherein said switching step is performed by system firmware.
Claim 11	
11	The method of claim 1 wherein the processor for which said LOL is detected includes a lockstep pair of processors, and wherein said switching comprises: determining one of the lockstep pair of processors that is not the cause of said LOL; and copying the state of the determined one of said lockstep pair of processors that is not the cause of said LOL to said spare processor.
Claim 12	
12	The method of claim 11 wherein said spare processor includes a lockstep pair of processors, and wherein said copying comprises: copying the state of the determined one of said lockstep pair of processors that is not the cause of said LOL to each of said lockstep pair of processors of said spare processor.

Claim 13	
[13pre]	A method comprising:
[13a]	system firmware determining that loss of lockstep (LOL) is detected for a lockstep pair of processors that are assigned the role of boot processor in a system;
[13b]	determining one of said lockstep pair of processors that is not the cause of said LOL;
[13c]	copying the state of the determined one of said lockstep pair of processors that is not the cause of said LOL to a spare processor;
[13d]	changing a system identifier, used by the system's operating system to recognize the boot processor, of said lockstep pair of processors that are assigned the role of boot processor, to a different value, which causes the system's operating system to not recognize the lock step pair of processors for which said LOL is detected as the boot processor; and
[13e]	changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to said spare processor.
Claim 14	
14	The method of claim 13 wherein the spare processor is a hot spare that is held in idle until switched the role of boot processor.
Claim 15	
15	The method of claim 13 wherein the spare processor is an active spare.
Claim 16	
16	The method of claim 15 further comprising: causing the system's operating system to idle the active spare.
Claim 17	
17	The method of claim 13 wherein said switching causes the system's operating system to recognize said spare processor as said boot processor without shutting down said operating system.
Claim 18	

18	The method of claim 13 wherein said determining step is performed by system firmware.
Claim 19	
19	The method of claim 13 wherein said copying step is performed by system firmware.
Claim 20	
20	The method of claim 13 wherein said switching step is performed by system firmware.
Claim 21	
[21pre]	A system comprising:
[21a]	a plurality of processor modules that each include a master processor and a slave processor that operate in lockstep;
[21b]	an operating system;
[21c]	error detection logic operable to detect loss of lockstep (LOL) for at least one of said processor modules; and
[21d]	system firmware operable, responsive to the detection of LOL for a first of said processor modules, to determine whether said first processor module is assigned a role of system boot processor;
[21e]	wherein if determined that said first processor module is assigned the role of system boot processor, said system firmware is further operable to cause said operating system to recognize another processor module as system boot processor without shutting down said operating system.
Claim 22	
22	The system of claim 21 wherein the another processor module is a processor module assigned a role of spare processor.
Claim 23	
23	The system of claim 21 wherein the another processor module is held in idle until said system firmware causes said operating system to recognize said another processor module as said system boot processor.
Claim 24	

24	The system of claim 21 wherein the system firmware causes the operating system to idle the another processor module before causing said operating system to recognize said another processor module as said system boot processor.
Claim 25	
25	The system of claim 21 further comprising: non-volatile data storage storing information that assigns the role of system boot processor to one of said plurality of processor modules.
Claim 26	
[26pre]	A system comprising:
[26a]	system firmware means for detecting loss of lockstep (LOL) for a processor in a multi-processor system;
[26b]	system software means for determining whether the processor for which said LOL is detected is assigned the role of boot processor; and
[26c]	system firmware means for switching the role of boot processor to a spare processor without shutting down the system's operating system if determined that the processor for which said LOL is detected is assigned the role of boot processor.
Claim 27	
27	The system of claim 26 wherein said processor in said multi-processor system is an Itanium Processor Family (IPF) processor.

I. Introduction

Patent Owner respectfully requests that the Board find all of Petitioner's challenges to the claims of U.S. Patent No. 7, 624, 302 ("the '302 patent") without merit. First, Petitioner's obviousness challenges, which are all based on the same combination of three prior art references and, for three of the four grounds, one or two additional references, fail to disclose, teach or suggest a limitation found in every challenged claim – switching the role of boot processor. Second, Petitioner's obviousness challenges also fail for claims 3, 4, 15, 16, and 24 because the prior art cited by Petitioner does not disclose, teach or suggest an "active spare" which is a limitation of these claims. Third, Petitioner's obviousness challenges also fail for claims 6, 8 and 25 because the prior art cited by Petitioner does not disclose, teach or suggest a "non-volatile data storage" which is a limitation of these claims. Fourth, Petitioner's obviousness challenges fail for claims 11, 12, 17 and 24 because Petitioner does not address these claims anywhere in the Petition. *See* Paper 1, "Pet." Nor are these claims addressed anywhere in Petitioner's Expert Declaration. *See* Ex. 1002, Declaration of Dr. Michael C. Brogioli ("Brogioli").

II. Background

The '302 Patent is titled "System and Method for Switching the Role of Boot Processor to a Spare Processor Responsive to Detection of Loss of Lockstep in a

Boot Processor” and was filed on October 25, 2004. *See* Ex. 1001. The '302 Patent was one of eight concurrently filed and commonly assigned patent applications, each of which was incorporated by reference in the '302 Patent. Ex. 1001, '302 Patent, 1:11-51.

The '302 Patent discloses a method comprising detecting loss of lockstep (LOL) for a processor in a multi-processor system, determining that the processor for which the LOL is detected is assigned the role of boot processor, and switching the role of boot processor to a spare processor without shutting down the system's operating system. *Id.*, Abstract.

The '302 Patent teaches that “[t]he traditional response to detected LOL has been to crash the system to ensure that the detected error is not propagated through the system.” *Id.*, 3:13-16. As noted in the '302 Patent, “crashing the system each time LOL is detected is not an attractive proposition,” especially for large systems having many processors. *Id.*, 3:19-25. The '302 Patent solves this problem by disclosing and claiming techniques for recovering from LOL detected for a boot processor in a multi-processor system, in which a spare processor is utilized for such recovery by assuming the role of boot processor. *Id.*, 4:60-5:5.

The '302 Patent, in conjunction with one of the concurrently filed applications, describes certain embodiments of a multi-processor system where

information is stored in a “device tree” that “identifies the processor in the system that is the boot processor and identifies the processor that is the spare for the boot processor.” *Id.*, 5:40-44. The '302 Patent also teaches that the system's firmware can access such information, and “upon detecting LOL for the boot processor, determines the spare processor from the device tree and assigns the spare processor the role of boot processor.” *Id.* 5:44–48. The '302 Patent further teaches that “[t]he firmware then attempts to recover lockstep for the processor for which LOL was detected,” and “[i]f the recovery of lockstep is successful, the processor for which lockstep was recovered (i.e., the former boot processor) may be assigned the role of spare for the current boot processor.” *Id.*, 5:48–53. As explained in the '302 Patent, “the multi-processor system may recover from LOL detected for the boot processor without requiring a crash of the system (at least for certain types of ‘recoverable’ LOLs).” *Id.*, 5:53-56.

The '302 Patent teaches that the spare processor for the boot processor may be a “hot” spare processor, which “is held in reserve, idling until such time as it assumes the role of boot processor responsive to detected LOL for the boot processor.” *Id.*, 5:57-63. The '302 Patent also teaches that the spare processor may be an “active” processor that is not held in reserve, but “is made available to the OS for executing instructions during normal system operation,” and “[u]pon LOL being

detected for the boot processor, [it] is dynamically transformed into a hot spare (i.e., it is idled) such that it can assume the role of the boot processor.” *Id.*, 5:63-6:2. Exemplary operational flow diagrams for switching the role of boot processor to a hot spare processor is shown in Figure 4, and for switching the role of boot processor to an active spare processor is shown in Figure 5 reproduced below.

FIG. 4

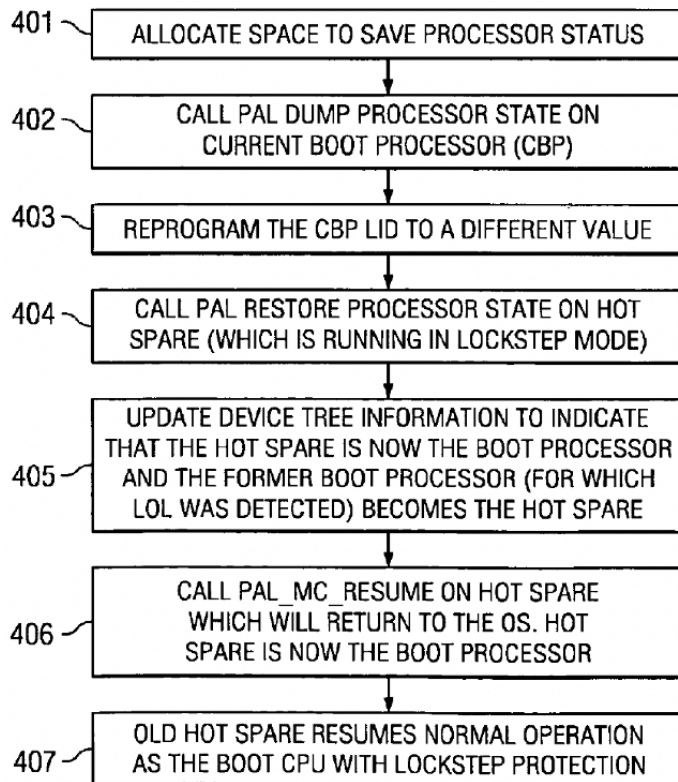
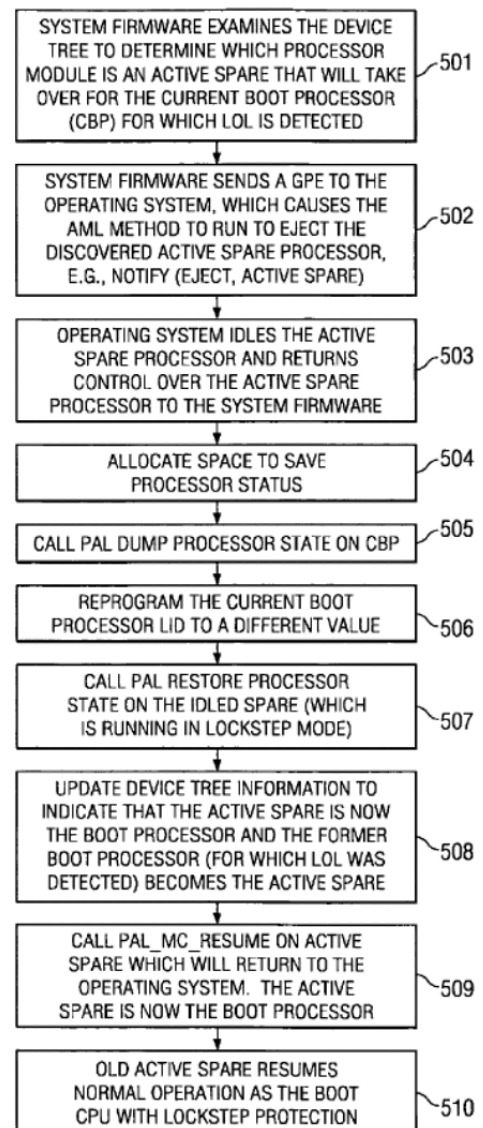


FIG. 5



The '302 Patent teaches that persistent store configuration data stored in non-volatile memory can be accessed, *id.*, 8:34-57, and information identifying the boot processor or a spare processor in non-volatile memory is further disclosed in, *e.g.*, “concurrently filed and commonly assigned U.S. patent application Ser. No. 10/973,077 (U.S. Published patent application 20060107112) titled “SYSTEM AND METHOD FOR ESTABLISHING A SPARE PROCESSOR FOR RECOVERING FROM LOSS OF LOCKSTEP IN A BOOT PROCESSOR, the disclosure of which is incorporated herein by reference, ...” *Id.*, 5:8-14, 34-40. This incorporated by reference patent application, which issued as U.S. Patent No. 7,627,781, teaches storing in NVRAM. *See, e.g.*, Ex. 2006, US 2006/0107112 A1, [0052].

The '302 Patent's inventions “enable the role of boot processor to be dynamically switched, during system runtime, to a spare processor responsive to a detected LOL for the boot processor, thus enabling recovery from the LOL without requiring that the system be crashed or re-booted.” *Id.*, 18:65-19:3.

III. Burden of Proof

To prevail in an *inter partes* review, Petitioner must show by a preponderance of the evidence that a patent is invalid. 35 U.S.C. § 316(e). “In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d

1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)); *see also Intelligent Bio-Systems, Inc. v. Illumina Cambridge Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016) (“It is of the utmost importance that petitioners in the IPR proceedings adhere to the requirement that the initial petition identify with particularity the ‘evidence that supports the grounds for the challenge to each claim.’”) (quotations omitted). This burden of persuasion never shifts to Patent Owner. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review). “Unlike district court litigation—where parties have greater freedom to revise and develop their arguments over time and in response to newly discovered material—the expedited nature of IPRs bring with it an obligation for petitioners to make their case in their petition to institute. While the Board’s requirements are strict ones, they are requirements of which petitioners are aware when they seek to institute an IPR.” *Intelligent Bio-Systems*, 821 F.3d at 1369.

Given the importance of particularity, the *inter partes* review rules set out in 37 C.F.R. § 42.104(b) specify the precise requirements for a petition to seek relief. In particular, the Petitioner must provide a statement of the precise relief requested

for each claim challenged as set forth in 37 C.F.R § 42.104(b) partially reproduced below.

(b) *Identification of challenge.* Provide a statement of the precise relief requested for each claim challenged. The statement must identify the following:

(1) The claim;

(2) The specific statutory grounds under 35 U.S.C. 102 or 103 on which the challenge to the claim is based and the patents or printed publications relied upon for each ground;

...

(4) How the construed claim is unpatentable under the statutory grounds identified in paragraph (b)(2) of this section. The petition must specify where each element of the claim is found in the prior art patents or printed publications relied upon; and

(5) The exhibit number of the supporting evidence relied upon to support the challenge and the relevance of the evidence to the challenge raised, including identifying specific portions of the evidence that support the challenge. The Board may exclude or give no weight to the evidence where a party has failed to state its relevance or to identify specific portions of the evidence that support the challenge.

37 C.F.R § 42.104(b) (2012).

IV. Person of Ordinary Skill in the Art

A person of ordinary skill in the art in the field of the '302 Patent would have been a person with at least a bachelor's degree in Electrical Engineering, Computer

Engineering, or equivalent, and at least two years of prior work experience with computer architecture, including experience relating to fault tolerant computer architecture, as of the earliest priority date of the '302 Patent—October 25, 2004. If a person did not have a bachelor's degree, additional relevant and practical experience with fault tolerant computer systems could provide an acceptable substitute. Likewise, further education beyond a bachelor's degree could compensate for a deficiency in practical experience. *See* Ex. 2004, Declaration of Nigel A. Jones ("Jones"), ¶ 26. This is similar, but not identical to, Petitioner's asserted person of ordinary skill in the art. *See, e.g.*, Paper 1, 4. Patent Owner contends that none of the prior art cited by Petitioner would have rendered any of the claims of the '302 Patent obvious to a POSITA under either Petitioner's or Patent Owner's proposed determination of the level of ordinary skill in the art and the analysis is identical for both proposed determinations.

V. Statement of Facts

Pursuant to 37 C.F.R. § 42.23(a), Patent Owner addresses Petitioner's statement of facts.

1. On or before October 25, 2004, using lockstep processors in a multiprocessor system was known. Admitted.
2. On or before October 25, 2004, it was known that a pair of lockstep

processors are often differentiated by designating one as a master processor and the other as the slave processor. Denied. *See* Ex. 1011, U.S. Patent No. 5,408,647 (“Landry”), 1:23-33 (“Asymmetrical multiprocessor systems, in which one microprocessor is the master and another microprocessor performs specific functions as a slave of the master microprocessor, are common and well known. Although the master/slave relationship improves computer performance due to the division of tasks, the computer does not operate at maximum capability. This is because the slave processor performs only particularly designated operations, and thus remains idle when a task not designated for the slave processor is performed.”).

3. On or before October 25, 2004, it was known that a multiprocessor system includes a boot processor that performs system initialization. Denied. *See, e.g.,* Ex. 1014, US 2003/0233492 A1 (“Schelling”), [0003] (characterizing selection of a boot strap processor as just “a common approach.”); Ex. 1023, U.S. Patent No. 6,081,890 (“Datta”), 1:28-31 (noting that a single bootstrap processor is only “usually” selected in multiprocessor systems).

4. On or before October 25, 2004, it was known that a multiprocessor system configured with IA-32 architecture includes a boot processor. Cannot be admitted or denied. The prior art cited by Petitioner, Ex. 1005, US 2004/0172164 A1 (“Fox”), does not mention a boot processor.

5. On or before October 25, 2004, it was known that a boot processor of a multiprocessor system could be identified as logical CPU0. Admitted.

6. On or before October 25, 2004, it was known to use a spare processor to recover a failing processor in a multiprocessor. Denied. The prior art cited by Petitioner discloses using a spare processor to replace a failed processor or recovering from a system error, but does not disclose recovering a failed processor.

7. On or before October 25, 2004, using both lockstep processors and spare processors in a multiprocessor system was known. Cannot be admitted or denied. The IBM S/390 Mainframe computer referenced by Petitioner may not have been what a POSITA considered a multiprocessor system. *See, e.g.*, Ex. 1005, Fox [0011] (noting that S/390 solutions are not available to the IA-32 architecture).

8. On or before October 25, 2004, using firmware to switch the role of a failing processor to a spare processor during system runtime was known. Denied. The prior art cited by Petitioner does not discuss switching roles.

9. On or before October 25, 2004, it was known that the BIOS of a computer system is firmware typically embedded in a non-volatile memory. Cannot be admitted or denied as “typically” is vague, ambiguous and undefined.

VI. The Challenged Claims of the '302 Patent are Not Unpatentable.

The Petition purports to challenge all of the claims of the '302 Patent on four different grounds. Ground 1 challenges claims 1-25 as obvious over the combination of Ex. 1005, Patent Application Publication US 2005/0172164 A1 ("Fox"), Ex. 1006, Patent Application Publication US 2004/0006722 A1 ("Safford"), and Ex. 1007, Patent Application Publication US 2006/0036889 A1 ("Arai"). Ground 2 challenges claims 1-25 as obvious over the combination of Fox, Safford, Arai and Ex. 1011, U.S. Patent No. 5,408,647 ("Landry"). Ground 3 challenges claims 26 and 27 as obvious over the combination of Fox, Safford, Arai, and Ex. 1012, Patent Application Publication US 2003/0126498 A1 ("Bigbee"). Ground 4 challenges claims 26 and 27 as obvious over the combination of Fox, Safford, Arai, Landry and Bigbee.

The '302 Patent has four independent claims – claims 1, 13, 21 and 25. The Petition generally refers back to the arguments set forth with respect to claim 1 for the limitations of claims 13 and 21. This is particularly true for the limitations reciting "transferring the role of" or "switching the role of" or "recognize another processor module as" boot processor. *See* Paper 1, Pet., 43 ("[13e] is similar to [1d]. Fox-Safford-Arai or Fox-Safford-Landry-Arai renders obvious [13e] as explained for Grounds 1-2, [1d]."), 52 ("As explained for Grounds 1-2, [1c]-[1d], in Fox-

Safford-Arai or Fox-Safford- Landry-Arai, the logical identifications (e.g., logical APIC IDs) of the lockstep processor pair of the boot processor detected to have LOL and the spare processor pair would have been switched.”).

Ground 1 fails because Fox in view of Safford and Arai would not have rendered claim limitation 1[d], “changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to a spare processor without shutting down the system's operating system,” obvious to a POSITA in 2004. Petitioner has failed to meet its burden of proof to establish that switching or transferring the role of boot processor would have been obvious to a POSITA from the combination of Fox in view of Safford and Arai. The role of boot processor applies during system initialization, and, as discussed below, Petitioner’s proposed combination would not result in the transferred or switched processor fulfilling the boot processor role during initialization. Ground 2 fails for the same reasons that Ground 1 fails, as well as the additional reasons set forth below.

Grounds 3 and 4 fail for similar reasons. Petitioner relies on its arguments with respect to claim limitation [1d] for claim limitation [26c]. *See* Pet., 73 (“As explained for Grounds 1-2, [1d], claims 10 and 20, Fox-Safford-Arai (Ground 1) or Fox-Safford-Landry-Arai (Ground 2) renders obvious system firmware (e.g., BIOS) for switching the role of boot processor to a spare processor without shutting down

the system's operating system.”). Because Petitioner's arguments have been refuted for Grounds 1 and 2, as discussed below, Petitioner's reliance on refuted grounds fails to meet Petitioner's burden of proof with respect to Grounds 3 and 4 and claims 26 and 27.

A. Grounds 1 and 2 do not establish that challenged dependent claims are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.

Because Grounds 1 and 2 fail for the independent claims, they fail for the dependent claims since a dependent claim cannot be obvious if the independent claim from which it depends is not obvious. Petitioner has therefore not established that dependent claims 2-12, 14-20, and 22-25 are unpatentable. *See, e.g., Comaper Corp. v. Antec, Inc.*, 596 F.3d 1343 (Fed. Cir. 2010). If an independent claim is nonobvious, then the dependent claim is also nonobvious. *In re Fine*, 837 F.2d 1071, 1076 (Fed. Cir. 1988) (“A claim that depends from a non-obvious independent claim is non-obvious because it contains all the limitations of the independent claim, plus a further limitation.”). The Board does not need to consider “arguments that certain dependent claim limitations would have been obvious where the base claim has not been proven invalid.” *SynQor, Inc. v. Artesyn Techs., Inc.*, 709 F.3d 1365, 1375 (Fed. Cir. 2013). *See also Zimmer Surgical, Inc. v Stryker Corp.*, 365 F.Supp. 3d 466, 487-488 (D. Del. 2019) (“As a dependent claim contains at least one more limitation than

the independent claim upon which it depends, it cannot be invalid under § 102 . . . if the independent claim upon which it depends is not anticipated.”) (*citing CA, Inc. v. Simple.com, Inc.*, 780 F. Supp. 2d 196, 260 (E.D.N.Y. 2009), *Duhn Oil Tool, Inc. v. Cooper Cameron Corp.*, 2012 U.S. Dist. LEXIS 24088, 2012 WL 604138 (E.D. Cal. Feb. 23, 2012)).

Petitioner's obviousness challenges in Grounds 1 and 2 also fail for dependent claims 3, 4, 15, 16, and 24 because Petitioner has not met its burden of establishing that the “active spare” limitation of these dependent claims would have been obvious to a POSITA as discussed below. Petitioner's arguments that an “active” spare would have been obvious to a POSITA from Fox are purely conclusory and significantly undermined by Dr. Brogioli's apparent misunderstanding of how an “active” spare differs from a “hot” spare as discussed below.

Petitioner's obviousness challenges in Grounds 1 and 2 also fail for dependent claims 6, 8 and 25 because the prior art cited by Petitioner does not disclose, teach or suggest the “non-volatile data storage” limitation of these claims as discussed below.

Finally, Petitioner's obviousness challenges in Grounds 1 and 2 fail for dependent claims 11, 12, 17 and 24 because Petitioner does not address these claims anywhere in the Petition and provides no explanation or rationale why these

dependent claims would have been obvious. These are not substitute claims raised in a motion to amend, and there is no basis for the Board to *sua sponte* identify patentability issues for these claims.

B. Ground 1 does not establish that challenged independent claims 1, 13 and 21 are rendered obvious by Fox in view of Safford and Arai.

As noted above, Petitioner asserts that independent claims 1, 13 and 21 would have been obvious to a POSITA over three references – Fox, Safford and Arai. Paper 1, Pet., 9-30, 36-53.

1. The Prior Art in Ground 1.

Fox is titled “Autonomous Fail-Over to Hot-Spare Processor Using SMI.” Ex. 1005, Fox. Fox is directed to “a method and system for dynamically activating a spare processor when processor failure is detected in a multi-processor data processing system.” *Id.*, [0002]. Fox describes a method and system for dynamically replacing a failing processor using Systems Management Interrupt (SMI) functionality provided within basic input output system (BIOS) of the IA-32 architecture. *Id.*, [0015]. At least one processor of the multiprocessor system (MP) is initially provided as a reserve (or hot-spare) processor that remains in an idle, off, or low-power mode. *Id.* While in that mode, the OS is prevented from initially utilizing the hot-spare processor. *Id.* When a processor failure is detected, SMI code running on a good processor instructs the OS to hold off allocating processes to the failing processor. *Id.*

Contemporaneously, the SMI (and OS) activates and completes an initialization of the hot-spare processor to prepare it to begin receiving the held-off processes. *Id.* Control is then returned to the OS, which updates the “active” processor list and allocates the threads that were running on the failing processor to the hot-spare processor. *Id.*

Fox does not disclose, teach or suggest a boot processor. Nor does Fox disclose, teach or suggest a BSP flag. Petitioner attempts to remedy some of these gaps in Fox's disclosure with an entirely different document that Petitioner did not identify in any of the asserted Grounds, Ex. 1008, IA-32 Intel® Architecture Software Developer's Manual, Vol. 3: System Programming Guide, (“IA-32”). Paper 1, Pet., 12-17. Petitioner attempts to remedy other gaps in Fox's disclosure with yet another entirely different document that Petitioner did not identify in any of the asserted Grounds, Ex. 1028, MultiProcessor Specification (“MP Specification”), Version 1.4. Paper 1, Pet., 13-17. As discussed below, neither of these documents have been authenticated and this Board should not consider these documents when evaluating the patentability of the claims of the '302 Patent. Both Exhibit 1008 and Exhibit 1028 are objectionable under Federal Rules of Evidence 802 and 901 because (1) the entirety of each exhibit is inadmissible hearsay to the extent it is offered for the truth of the matter asserted, and (2) Petitioner has not produced

evidence sufficient to support a finding that either exhibit is what Petitioner claims it is.

Safford is titled “Method and Apparatus for Recovery from Loss of Lock Step.” Ex. 1006, Safford. Safford describes a computer system using lock-stepped processor cores that operate in a master/checker pair to improve reliability of processing assets. *Id.*, [0015]. Each of the processors in the pair processes the same code sequences. *Id.* When a logic circuit compares the resulting outputs of the processors, any difference in the processor outputs indicates the existence of an error, prompting the logic circuit to initiate a sequence of steps to halt operation of the two processors. *Id.*

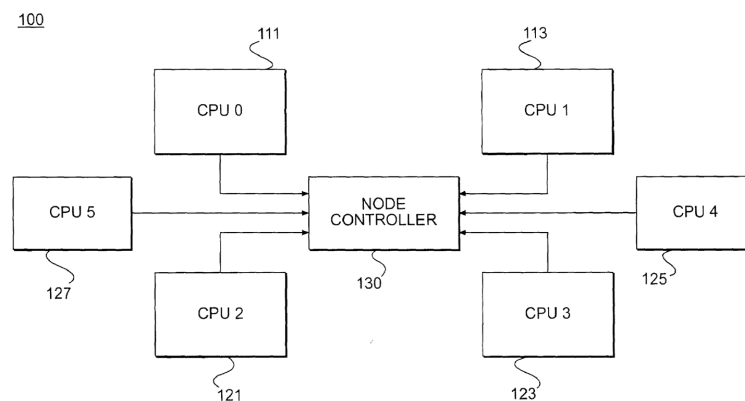


FIG. 3

Figure 3 of Safford, reproduced above, depicts computer system 100 with processors 111, 113, 121, 123, 125, and 127 that operate as pairs when in lock step

(i.e., processors 111 and 113 are a first pair; processors 121 and 123 are a second pair; and processors 125 and 127 are a third pair). Ex. 1006, Safford, [0018]. The processor pairs are coupled to a lockstep logic, connected to node controller 130. *Id.*

Safford teaches that “[t]he processor 125 may be designated as a ‘hot standby,’ and is sitting idle in lock step mode with the processor 127.” *Id.*, [0020]. “Should one of the processors 111, 113, 121, and 123 suffer an error, the hot standby processors 125, 127 may be used to speed recovery from the resulting loss of lock step.” *Id.*

Arai is titled “High Availability Multi-Processor System.” Ex. 1007. Arai describes “a method and system for reserving a processor in a multiprocessor system for availability of replacement in response to detection of an error in a non-reserved processor.” *Id.*, [0002]. Arai teaches that “[w]hen a central processor in a multiprocessor system encounters an error, it is very desirable to not lose the work being done on that processor and to move that work to another processor that is still operating in the system.” *Id.*, [0005].

In Arai, during system initialization, one or more logical processors may be reserved in an inactive state. *Id.*, Abstract. In the event an error is detected on a logical or physical processor, the execution context of the processor experiencing the error is transferred to one or more of the reserved logical processors. *Id.* Thereafter, the active processor is designated as inactive and replaced by the inactive processor to which the execution context has been transferred. *Id.*

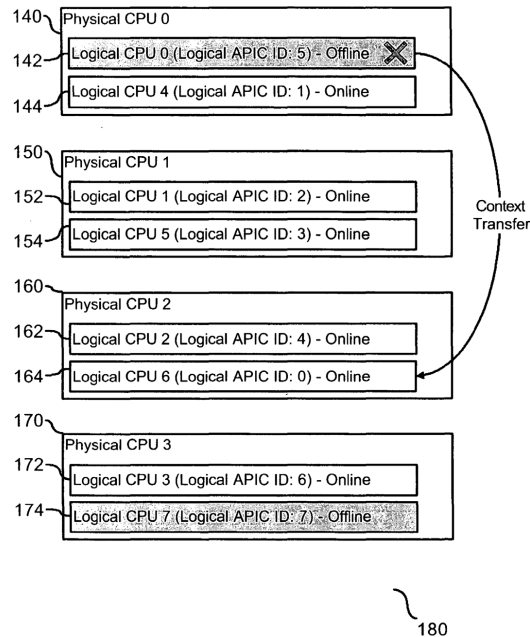


FIG. 4b

Figure 4b of Arai, reproduced above, depicts the identification and status of each processor following occurrence of an error in a logical processor. *Id.*, [0016]. Following detection of an error with logical CPU0 (142), logical CPU0 is placed in an idle state and assigned the identification number of logical CPU6 (164). *Id.*, [0023].

At the same time, logical CPU6 (164) is placed in an active state and assigned the identification number of CPU0 (142). *Id.* In this way, in response to an error in an active processor, the execution context of the active processor may be transferred to an identified inactive processor, followed by activation of the processor that has received the execution context. *Id.*, [0025].

2. Petitioner has not met its burden to establish claim 1 is unpatentable.

Petitioner's proposed combination of these three references in the Petition does not render claim 1 of the '302 Patent obvious. The primary reference, Fox, does not disclose a boot processor. Ex. 2004, Jones, ¶ 28. Nor does Fox disclose, teach or suggest a BSP flag which is what Petitioner asserts designates a boot processor. Paper 1, Pet., 16-18. Petitioner attempts to remedy some of these gaps in Fox's disclosure with an entirely different document that Petitioner did not identify in any of the asserted Grounds, Ex. 1008, IA-32 Intel® Architecture Software Developer's Manual, Vol. 3: System Programming Guide, ("IA-32"). Paper 1, Pet., 12-17. Petitioner attempts to remedy other gaps in Fox's disclosure with yet another entirely different document that Petitioner did not identify in any of the asserted Grounds, Ex. 1028, MultiProcessor Specification ("MP Specification"), Version 1.4. Paper 1, Pet., 13-17.

Neither of these exhibits have been authenticated and this Board should not consider these documents when evaluating the patentability of the claims of the '302 Patent. Both Exhibit 1008 and Exhibit 1028 are objectionable under Federal Rules of Evidence 802 and 901 because (1) the entirety of each exhibit is inadmissible hearsay to the extent it is offered for the truth of the matter asserted, and (2) Petitioner has not produced evidence sufficient to support a finding that either exhibit is what Petitioner claims it is.

As Petitioner's expert acknowledged, Fox does not disclose a boot processor or the BSP flag Petitioner relies upon for designating a boot processor. Paper 1, Pet., 16-18. The only evidence in the record of a BSP flag comes from Exhibit 1008, which Petitioner refers to as "IA-32." However, IA-32 is a nearly 800-page document that has not been shown to be prior art. When Petitioner's expert, Dr. Brogioli, was asked during deposition where he obtained Exhibit 1008, he could not recall. Ex. 2007, Brogioli Dep. Tr., 27:6-28:14. When asked whether he did anything to ascertain whether Exhibit 1008 was publicly available as of the copyright date listed on the document, he could only describe how he "would have looked" for a document like this, and he could not confirm that he actually did any further investigation. Ex. 2007, Brogioli Dep. Tr., 30:5-31:24. Although Petitioner did include a declaration regarding the Internet Archive in the table of exhibits, Paper 1,

Pet., ix, that declaration is not cited anywhere else in the Petition. Nor should it have been. A cursory examination of Ex. 1009 reveals that it is directed to a web page from February 14, 2002, but Ex. 1008 bears an apparent copyright date of 2003. Ex. 1008, IA-32, 2. Accordingly, Ex. 1008 could not have been accessed from the web page referenced in Ex. 1009.

In addition, neither Petitioner nor Dr. Brogioli presents any evidence or argument that a POSITA would have been aware of Exhibit 1008 at the time of the invention of the '302 Patent or that the nearly 800-pages contained within Exhibit 1008 would have been part of the background knowledge in the art. Nor has Petitioner presented any evidence or argument that a POSITA reading Fox would have looked to Exhibit 1008 to understand any disclosure in Fox, other than the inclusion of the phrase "IA-32" in both Fox and Exhibit 1008. Dr. Brogioli has not presented evidence as to: whether Ex. 1008 was available to a POSITA at the time of the invention of the '302 patent, why a POSITA would have understood Ex. 1008 as a relevant description of the system described in Fox, or why any of the implementation details in Ex. 1008 would have been known in the art at the relevant time. *See* Ex. 2006, Jones, ¶ 31. For at least these reasons, the disclosures in Exhibit 1008 relating to a BSP flag are insufficient to demonstrate obviousness of either "determining that the processor for which said LOL is detected is assigned a role of

boot processor” or “changing a system identifier, used by the system’s operating system to recognize the boot processor” limitations at the time of the invention of the ’302 patent.

The Petition is similarly deficient with respect to Ex. 1028, MP Specification. The MP Specification is a multi-colored document in so much as some of the text is blue and some of the text is black. When Dr. Brogioli was asked “how do you interpret the fact that several of the entries in this table are written in blue font instead of black?” he responded that “it looks like this is the actively drafted version.” Ex. 2007, Brogioli Dep. Tr., 34:10-35:9. Dr. Brogioli’s speculation that the admitted draft version of Ex. 1028 was “made available” is uncorroborated and based on unsupported speculation regarding “various versions of these types of documents.” Ex. 2007, Brogioli Dep. Tr., 36:1-20. Petitioner presents no evidence that Ex. 1028 was ever available outside of Intel.

A copyright notice informs the public that copyright protection is claimed, identifies the copyright owner, and states the asserted year of first publication. 17 U.S. C. § 401. The purpose of a copyright notice is, simply, to put a reader on notice that a claim has been made that the work is copyrighted. It does not establish copyright protection and is not required. It also does not establish when a document

was publicly accessible under patent law. *Stryker Corp. v. Karl Storz Endoscopy America, Inc.*, IPR2015-00677, Paper 15, 18-19 (PTAB Sept. 2, 2015).

3. Petitioner has not met its burden to establish claim 1's limitation of switching the role of boot processor to a spare processor would have been obvious.

Notwithstanding the foregoing, even considering Ex. 1008 and Ex. 1028, the Petition does not demonstrate obviousness of the limitations of claim 1 of the '302 Patent reciting “determining that the processor for which said LOL is detected is assigned a role of boot processor” or “changing a system identifier, used by the system's operating system to recognize the boot processor” or “changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to a spare processor without shutting down the system's operating system.” limitations at the time of the invention of the '302 patent.

The designation of a boot processor (BSP) and applications processors (APs) in Ex. 1008 (referred to by Dr. Brogioli as “IA-32”) is disclosed as being performed following a power-up or reset. Ex. 1008, 238 “Following a power-up or RESET of an MP system, system hardware dynamically selects one of the processors on the system bus as the BSP.” Ex. 2006, Jones, ¶ 29.

The Petition and Dr. Brogioli allege that in Ex. 1008, IA-32, and Ex. 1028, MP Specification, “the BSP executes the BIOS code to boot the operating system,

initialize the APICs, and build the MP configuration table to communicate to the operating system about the APICs and APs.” Ex. 1002 ¶ 47. The Petition and Dr. Brogioli also allege that “once the operating system is up and running, the BSP functions as an AP.” *Id.* (citing Ex. 1028 at 75). Ex. 2006, Jones, ¶ 30.

Petitioner and Dr. Brogioli allege “Fox would have designated one of its processors as the boot processor responsible for initializing the multiprocessor system and booting the operating system.” Pet. 15 (citing Ex. 1008 at 238-239). I disagree that this is how a POSITA would have understood Fox. Fox does not mention a boot processor, and there is no evidence that a POSITA would have implemented Fox in the manner described in Ex. 1008 at 238-239. Ex. 2006, Jones, ¶ 32.

Dr. Brogioli and the Petition are not incorrect that “initializing the multiprocessor system and booting the operating system” can be part of the role of boot processor. Ex. 2006, Jones, ¶ 33. But Petitioner and Dr. Brogioli are not correct when they contend that the role of boot processor is transferred to a spare processor in the Fox-Safford-Arai combination. *See* Paper 1, Pet., 27. “A POSITA would have understood that such switching would have changed the logical APIC ID (‘system identifier’) of the processor in error to the logical APIC ID of the reserved processor (‘a different value’). [Arai, [0020]-[0020], [0025]-[0026]]; Brogioli, ¶121.” Even if

that proposition is correct, it does not demonstrate the “role of boot processor” has been transferred to a spare processor. For example, Petitioner provides no evidence that changing the logical APIC ID of the processor in error to the logical APIC ID of the reserved processor has any impact on which processor performs any boot processor role, including for example the functions of initializing the system or booting the operating system. Indeed, Dr. Brogioli's contention that, in IA-32, the boot processor becomes an applications processor after the boot process is finished is at odds with his contention that transferring APIC IDs effectuates a transfer, during runtime, of the role of boot processor to a spare processor. Ex. 2006, Jones, ¶ 34.

Furthermore, the processor identifiers in Arai are created in a table in RAM during the boot process by the BIOS. Ex. 1007, Arai [0020] (“the system initialization includes generating a table in which each processor in the system is designated as active and available to the operating system or in an idle state and unavailable to the operating system”), *id.*, [0021] (“The BIOS creates a table in the main memory (RAM) during start-up”). Dr. Brogioli testified that Arai's table designating active/idle processors is stored in RAM. Ex. 2007, Brogioli Dep. Tr., 103:14-21. With respect to the Fox-Arai combination, Dr. Brogioli states that the register or table used in Fox for identifying each processor's operational state would

be stored in RAM, as it is in Arai. Ex. 2007, Brogioli Dep. Tr., 108:10:10-109:4; *see also Id.* at 110:4-11, 111:12-112:12. Ex. 2006, Jones, ¶ 35.

There is no evidence in Arai, or from Dr. Brogioli, that changes to the APIC ID play any role in boot processor functions. Indeed, Dr. Brogioli testified that, in Arai, “every time you power the machine up or reboot it – or power cycle it, I should say – the BIOS runs again out of this non-volatile memory that’s a ROM or an EEPROM, as an example.” Ex. 2007, Brogioli Dep. Tr., 111:7-11. A POSITA would readily recognize, based on Arai’s disclosure of how the table of APIC IDs is created by the BIOS during bootup and written to RAM, that changes to logical identifiers made during runtime have no impact on which processor assumes the role of boot processor. A POSITA would understand from Arai that because the table of APIC IDs is created by the BIOS during bootup, the APIC IDs themselves does not assign the role of boot processor, nor does swapping APIC IDs in Arai “transfer, during runtime, the role of boot processor to a spare processor.” An identifier, such as Arai’s APIC ID, stored in RAM would not persist through a reset. Ex. 2006, Jones, ¶ 36.

In contrast, the ’302 specification describes a different approach wherein, for example, information regarding the boot device path can be changed during runtime and that is then persistently stored “for the next time the system reboots.” Ex. 1001 at 12:56; *see also id.* at 12:36-59, 8:34-59. Ex. 2006, Jones, ¶ 37.

Dr. Brogioli also says that Arai teaches “transfer[ring] the execution context of the failing processor to the reserved processor.” Ex. 1002, Brogioli, ¶ 50 (citing Arai [0002]-[0023], [0025], FIGS 4a-4b). As a POSITA would readily recognize, transferring the execution context of a failing processor to a reserved processor is different from transferring the role of boot processor to reserved processor. Execution context generally refers to the information a processor needs to execute a thread or process. Dr. Brogioli has identified no evidence in Arai that any aspect of the “execution context” plays any part in transferring the role of boot processor. Ex. 2006, Jones, ¶ 38.

Dr. Brogioli similarly states that "Fox teaches that during the SMI, the system state of each processor is completely saved to memory, the caches and pipelines are flushed, and the BIOS switches the memory save state from the old processor (i.e., the failing processor) to the new processor (i.e., the spare processor)." Ex. 1002, Brogioli, ¶ 105 (citing Fox, [0041]). A POSITA would not understand switching the memory save state from one processor to another, as described in Fox, to cause a transfer of the role of boot processor because (1) Fox does not disclose a boot processor and (2) even if one assumed that Fox had a boot processor, there is no evidence that the role of boot processor is designated using the memory save state. Ex. 2006, Jones, ¶ 39.

In paragraph 107 of Dr. Brogioli's declaration, he contends that "both Fox and Arai disclose or suggest . . . using firmware to designate the processor roles during system initialization, identify the failing processor, and replace the failing processor with the spare processor." In paragraph 121, he similarly states that "Arai discloses that during system initialization, the BIOS creates a table that identifies each processor in the system and its designated operational state, i.e., active or disabled. *Id.*, [0019]-[0022]. If the BIOS determines the error requires replacement of the processor, and if there is reserved inactive processor, the identification numbers of the active processor in error and the reserved inactive processor are switched."

To the extent that Dr. Brogioli's statements in his declaration were intended to imply that either Fox or Arai discloses or suggests designating processor roles during system initialization based on some prior identification of a failing processor and/or replacement of a failing processor with a spare processor, he is wrong. None of the cited portions of Fox (*e.g.*, [0040]-[0042]) or Arai (*e.g.*, [0020], [0022]-[0024]) provide any support for that contention because they do not teach or suggest a relationship between a failing processor and processor roles during system initialization. Ex. 2006, Jones, ¶ 39.

For at least these reasons, Fox in view of Safford and Arai do not render obvious limitation 1[d], i.e., "changing a system identifier of the spare processor to

that of the boot processor, thereby switching the role of boot processor to a spare processor without shutting down the system's operating system.” Ex. 2006, Jones, ¶ 40.

C. Ground 1 does not establish that challenged independent claims 13 and 21 are rendered obvious by Fox in view of Safford and Arai.

As noted above, Petitioner also asserts that independent claims 13 and 21 would have been obvious to a POSITA over the same three references – Fox, Safford, and Arai. Paper 1, Pet., 36-53.

For claim 13, Dr. Brogioli and the Petition assert that claim limitation [13e] “changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to said spare processor” is substantially similar to claim limitation 1[d] and would have been obvious for the same reasons set forth with respect to Grounds 1 and 2 and limitation [1d]. Ex. 1002, Brogioli ¶ 164, Paper 1, Pet., 43. Petitioner’s arguments fail for the same reasons they fail with respect to limitation [1d]. Ex. 2006, Jones, ¶ 48.

For claim 21, Dr. Brogioli and the Petition asserts that claim limitation [21e] “wherein if determined that said first processor module is assigned the role of system boot processor, said system firmware is further operable to cause said operating system to recognize another processor module as system boot processor without shutting down said operating system.” would have been obvious for the same

reasons set forth with respect to Grounds 1 and 2 and limitation [1d] and 13[a]. Ex. 1002, Brogioli, ¶ 185, Paper 1, Pet., 51-52.

A POSITA would understand that “caus[ing] said operating system to recognize another processor module as system boot processor” is equivalent to “changing the role of boot processor to a spare processor” consistent with the specification of the '302 Patent. The '302 Patent specification teaches “[e]xemplary techniques that may be utilized by firmware 18 for switching the **role** of boot processor to the hot spare.” Ex. 1001, 11:16-18 (emphasis added). Ex. 2006, Jones, ¶ 51.

Petitioner's arguments fail for the same reasons they fail with respect to limitation [1d] and for the reason stated above. Ex. 2006, Jones, ¶ 52.

D. Ground 2 does not establish that challenged independent claims 1, 13 and 21 are rendered obvious by Fox in view of Safford, Landry and Arai.

As noted above, Petitioner also asserts that independent claims 1, 13 and 21 would have been obvious to a POSITA over four references – Fox, Safford, Arai and Landry. Paper 1, Pet., 30-66.

1. The Additional Prior Art in Ground 2.

Landry is titled “Automatic Logical CPU Assignment of Physical CPUs.” Ex. 1011, Landry. Landry relates to “startup logic for assigning logical central processing unit (CPU) designations among multiple CPUs.” *Id.*, 1:7-10. Landry discloses a multiprocessor computer system that includes fault tolerant power up logic for finding a functioning CPU to operate as logical CPU0. *Id.*, 2:9-11. When the system is powered up, all of the CPUs except the CPU in physical slot 0 (CPU P0) are initially placed in a SLEEP mode. *Id.*, 2:15-18. The microprocessor in physical location 0 performs its power on self-test (POST) routine and tests its associated cache memory system. *Id.*, 2:18-20. If the CPU functions properly, the CPU is designated as CPU L0, and the microprocessor retains this designation until the system is power cycled. *Id.*, 2:20-23. The CPU designated as CPU L0 then awakens the remaining CPUs and boots up the rest of the computer system. *Id.*, 2:23-25.

If CPU P0 is not functioning properly, it is designated as inoperative. *Id.*, 2:26-27. The system then awakens CPU P1 and repeats the process of testing the CPU. If CPU P1 is operational, then it is designated as CPU L0, and it boots the remainder of the system. *Id.*, 2:27-31. On the other hand, if CPU P1 also fails, it is also given an inoperative designation. The computer system then turns on CPU P2, and repeats the process. *Id.*, 2:31-32. The process repeats until an operational microprocessor is found to perform the CPU L0 functions. *Id.*, 2:33-35.

2. Petitioner has not met its burden to establish claim 1 is unpatentable in Ground 2.

Petitioner and Dr. Brogioli contend, with respect to claim limitation 1[b] (“determining that the processor for which said LOL is detected is assigned a role of boot processor;”), that “Fox in view of Safford and Landry further renders obvious determining that the processor for which said LOL is detected is assigned a role of boot processor. Ex. 1002, Brogioli, ¶ 141, Paper 1, Pet., 33-34. Dr. Brogioli and Petitioner further contend that “a POSITA would have understood that Fox’s system would have, or it would have been obvious to have, a boot processor, and would have implemented Landry’s boot processor selection process in Fox and identified logical CPU0 in Fox as the boot processor. Thus, Fox combined with Safford and Landry, would have determined that the processor for which said LOL is detected is logical CPU0, which is the processor assigned a role of boot processor.” Ex. 1002, Brogioli, ¶ 143, Paper 1, Pet., 35. Petitioner and Dr. Brogioli also contend, with respect to claim limitation 1[c] (“changing a system identifier, used by the system's operating system to recognize the boot processor, of said boot processor for which said LOL is detected, to a different value, which causes the system's operating system to not recognize the processor for which said LOL is detected as the boot processor; and”), that “[a] POSITA would have understood that in Fox combined with Safford, Landry, and Arai, the system identifier used by the operating system

to recognize the boot processor would be the identifier of logical CPU0.” Paper 1, Pet., 35, Ex. 1002, Brogioli, ¶ 144.

Importantly, with respect to claim limitation 1[d], Dr. Brogioli and Petitioner contend that “a POSITA would have understood that in Fox combined with Safford, Landry, and Arai, the system identifier of the spare processor would be changed to the identifier of logical CPU0 to replace the boot processor that has lost lockstep.” Paper 1, Pet., 36, Ex. 1002, Brogioli, ¶ 145. Petitioner's arguments fail for the same reasons they failed for this limitation in Ground 1. Ex. 2006, Jones, ¶ 43.

Dr. Brogioli and Petitioner have not shown that switching the logical CPU0 as taught by Safford and Arai have the effect of switching the role of boot processor to a spare processor. As Dr. Brogioli states “such switching only requires ordinary computer programming skills to assign unique logical identifications to processors in Fox's system and update Fox's processor register or table storing the processor's identifications, as known in the art.” Ex. 1002, Brogioli, ¶ 150. But, as previously explained, assigning logical identifications to processor and/or updating registers or tables storing processor identifications does not, by itself, teach or even suggest that the role of boot processor has been switched. Ex. 2006, Jones, ¶ 44.

To the extent that Dr. Brogioli and Petitioner intend to rely on Landry's disclosures relating to “boot processor selection,” none of those disclosures are

relevant to “switching the role of boot processor to a spare processor without shutting down the system’s operating system” because Landry’s boot processor selection happens during system initialization, *i.e.*, before the operating system is running. “Shutting down the operating system” can only occur if the operating system is running. Ex. 2006, Jones, ¶ 44. Indeed, as Dr. Brogioli contends in ¶ 132 of his declaration, “Landry’s process would have been a suitable option for selecting a boot processor in Fox’s system consistent with the initialization protocol of IA-32 microprocessor system.” Ex. 2006, Jones, ¶ 44.

3. Petitioner has not met its burden to establish claims 13 and 21 are unpatentable in Ground 2.

Petitioner does not provide separate arguments for claims 13 and 21 under Grounds 1 and 2.

For claim 13, Dr. Brogioli and the Petition assert that claim limitation [13e] “changing a system identifier of the spare processor to that of the boot processor, thereby switching the role of boot processor to said spare processor” is substantially similar to claim limitation 1[d] and would have been obvious for the same reasons set forth with respect to Grounds 1 and 2 and limitation [1d]. Ex. 1002, Brogioli ¶ 164, Paper 1, Pet., 43. Petitioner’s arguments fail for the same reasons they fail with respect to limitation [1d]. Ex. 2006, Jones, ¶ 48.

For claim 21, Dr. Brogioli and the Petition asserts that claim limitation [21e] “wherein if determined that said first processor module is assigned the role of system boot processor, said system firmware is further operable to cause said operating system to recognize another processor module as system boot processor without shutting down said operating system.” would have been obvious for the same reasons set forth with respect to Grounds 1 and 2 and limitation [1d] and 13[a]. Ex. 1002, Brogioli, ¶ 185, Paper 1, Pet., 51-52.

A POSITA would understand that “caus[ing] said operating system to recognize another processor module as system boot processor” is equivalent to “changing the role of boot processor to a spare processor” consistent with the specification of the '302 Patent. The '302 Patent specification teaches “[e]xemplary techniques that may be utilized by firmware 18 for switching the **role** of boot processor to the hot spare.” Ex. 1001, 11:16-18 (emphasis added). Ex. 2006, Jones, ¶ 51.

Petitioner's arguments for claims 13 and 21 fail for the same reasons they fail with respect to limitation [1d] and for the reason stated above. Ex. 2006, Jones, ¶ 52. For the reasons set forth above, Petitioner fails to meet its burden to establish that claims 1, 13, and 21 are unpatentable in Ground 2.

E. Grounds 3 and 4 do not establish that challenged independent claim 26 or dependent claim 27 are rendered obvious by Fox in view of Safford, Arai and Bigbee or Fox in view of Safford, Landry, Arai, and Bigbee.

As noted above, Petitioner asserts that independent claim 26 and dependent claim 27 would have been obvious to a POSITA over a combination of four or five references – Fox, Safford, Arai, and Bigbee or Fox, Safford, Arai, Bigbee, and Landry. Paper 1, Pet., 67-74.

Bigbee is titled “Method and Apparatus for Functional Redundancy Check Mode Recovery.” Ex. 1012, Bigbee. Bigbee describes data processing systems where “two or more physical processing elements or ‘cores’ are associated with and function as a single logical processor.” *Id.*, [0002]. In Bigbee’s disclosure, “[t]he processor cores are operated in a functional redundancy check (FRC) mode in which identical instructions are provided to each core and concurrently executed on identical data so that any error occurring within a single core will produce an inconsistent result as compared to the remaining cores associated with a given processor.” *Id.* Specialized FRC logic performs this comparison and generates a signal such as an interrupt or exception once an inconsistent result has been detected. *Id.*

Petitioner and Dr. Brogioli assert that claim limitation [26c] (“system firmware means for switching the role of boot processor to a spare processor without

shutting down the system's operating system if determined that the processor for which said LOL is detected is assigned the role of boot processor”) is rendered obvious for, *inter alia*, the same reasons applied to limitation [1d] in Grounds 1 and 2. Paper 1, Pet., 73, Ex. 1002, Brogioli, ¶ 251.

Petitioner and Dr. Brogioli have not met their burden with respect to claim 26 for the same reasons that they did not meet their burden with respect to claim 1. Ex. 2006, Jones, ¶ 55.

F. Grounds 1 and 2 do not establish that challenged dependent claims 3, 4, 15, 16, and 24 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.

As noted above, Petitioner asserts that dependent claims 3, 4, 15, 16 and 24 would have been obvious to a POSITA over a combination of three or four references – (1) Fox, Safford, and Arai or (2) Fox, Safford, Arai and Landry. Paper 1, Pet., 56-58. As discussed below, Petitioner does not present any argument or evidence with respect to claim 24 and thus cannot meet its burden of proof to establish unpatentability of claim 24.

Petitioner's arguments fail because claims 3, 4, 15 and 16 recite limitations requiring an active spare. Claim 3 recites “the spare processor is an active spare.” Ex. 1001, '302 Patent, 19:24-25 (emphasis added). Claim 4, which depends from claim 3, recites “causing the system's operating system to idle the active spare.” *Id.*,

19:28-29. Claim 15 recites “the spare processor is an active spare.” *Id.*, 20:23-24 (emphasis added). Claim 16, which depends from claim 15, recites “causing the system's operating system to idle the active spare.” *Id.*, 20:27-28. The '302 Patent discloses that, as an alternative embodiment, “a processor can be dynamically made a ‘spare’ when needed for recovery from LOL encountered for the system boot processor.” *Id.*, 15:24-27. The '302 Patent teaches that “the firmware may send an ‘eject request’ on a healthy processor pair (e.g., CPU 3), and lie to the OS by indicating (e.g., in response to an ACPI_STA method for the healthy processor pair) that such processor pair is not functioning. This would cause the OS to eject that processor pair as though it had encountered a LOL. This processor would become the ‘hot spare’ and be used to replace the boot processor.” *Id.*, 15:43-50.

The '302 Patent also discloses using an active spare processor in connection with Figure 5. The process for using an active spare is more complex and time-consuming than the process for using a hot spare. *Compare* Figures 4 and 5, reproduced below.

FIG. 4

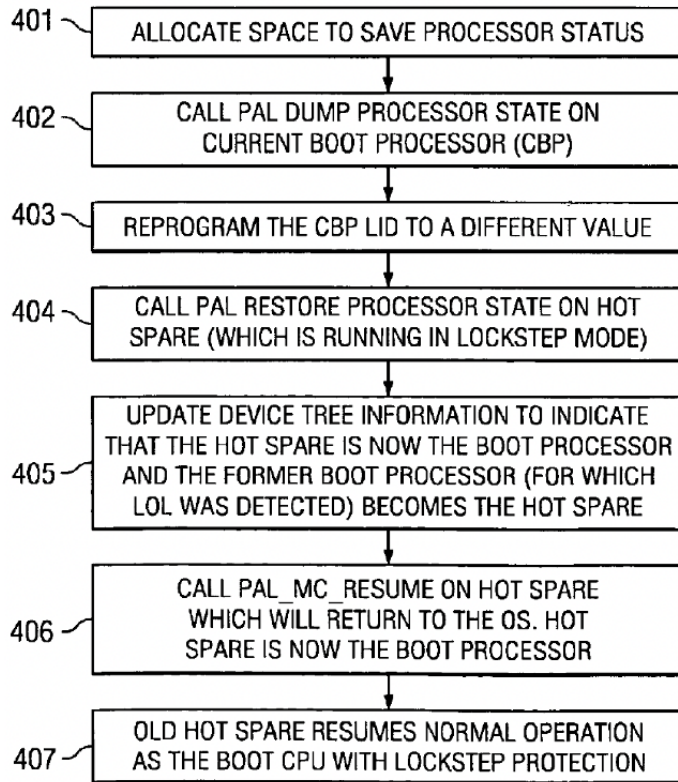
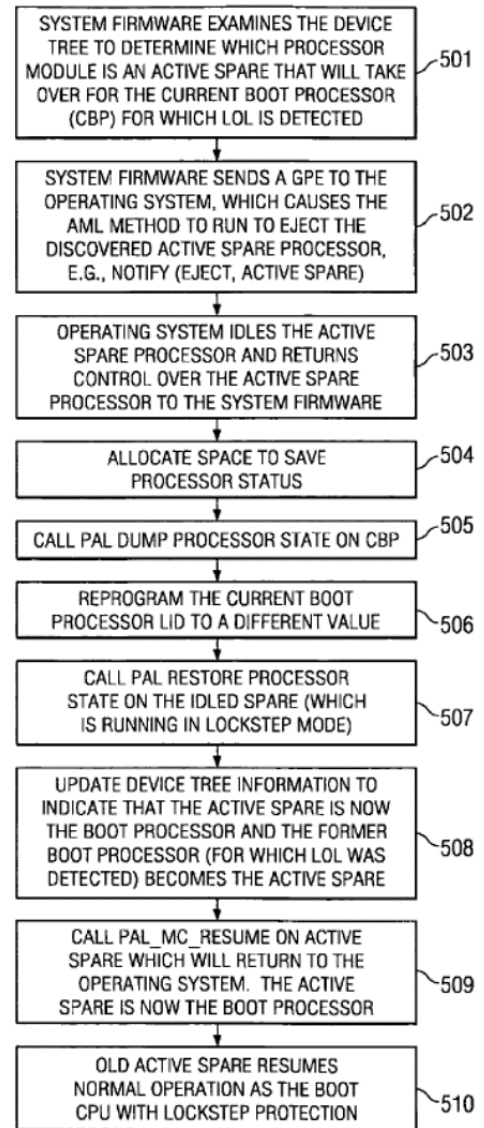


FIG. 5



As can be seen from Figure 5, using an active spare requires three additional steps 501, 502, and 503. The '302 Patent teaches that the following actions are required. "In operational block **501**, the system firmware, responsive to such detection of LOL for the boot processor, examines the device tree to determine which processor module is assigned the role of active spare. In operational

block **502** the system firmware (e.g., SAL) asserts a General Purpose Event Interrupt (GPE) by writing to the appropriate GPE register in the ACPI register space. This causes the OS to execute its interrupt handler, which will run. an AML Method associated with this particular interrupt. In this example, this AML method that is executed responsive to the interrupt generated by the GPE includes an AML instruction called “Notify”. This “Notify” operation indicates the determined active spare processor object and that an Eject is being requested as arguments. Accordingly, an AML method is used to request the OS to eject the active spare that was determined in block **501**. In response, the OS idles the active spare processor and returns control over the active spare processor to the system firmware, in block **503**.” Ex. 1001, '302 Patent, 17:12-32.

The Petitioner relies on the disclosure of Ex. 1005, Fox, for the active spare limitations. Paper 1, Pet., 56-58. Specifically, Petitioner relies on an unexplained alternate embodiment that does not explicitly disclose or teach using an active spare. Petitioner contends that an active spare is disclosed by Fox's statement that “in another embodiment[,], the system is made to enable the ‘hot spare processor’ feature on-the-fly by the OS with a driver-OS-SMI handshake. Tracking which processors are active may be completed by [a] software table of active processors, which is

available to be read by the OS, when determining workload allocation to each of the active processors.” Paper 1, Pet., 56.

Petitioner's reading of Fox cannot be reconciled with the entirety of Fox's disclosure. Petitioner is apparently arguing that a hot-spare processor and an active spare processor are the same as Dr. Brogioli testified.

4 Q. So are you saying that that
5 processor is still a hot-spare processor and
6 now that it's been added to the active list
7 is also an active spare?

8 A. That seems to me the language
9 that Fox is using.

10 Q. Is that your understanding of
11 Fox?

12 A. Yeah. That's what he is saying
13 and that is why these sequences of
14 paragraphs are what show both -- taking us
15 back to my declaration -- what is both
16 dependent claim 13 and dependent claim 14.

Ex. 2007, Brogioli Dep. Tr., 78:4-16.

However, that is not how a POSITA would understand Fox's disclosure. As an initial matter, Fox is titled “Autonomous Fail-Over to Hot-Spare Processor Using SMI.” Ex. 1005, Fox. An asserted benefit of Fox is that “[n]egligible system

downtime occurs during the transfer of executing threads/processes to the replacement processor, unlike other methods” *Id.*, [0014]. Fox further asserts that “a processor is able to autonomically fail-over to a hot-spare processor without affecting the OS or executing applications” and “the invention provides a method for dynamic, hot swap of spare processors...”. *Id.*, [0016]. To achieve these goals, Fox teaches that “[w]hen the MP is initially activated for processing, one IA-32 processor is held off as a spare processor. Holding off the processor involves having the basic input output system (BIOS) recognize the processor as a spare during BIOS boot (i.e., power on self test (POST), etc. of the MP and having the BIOS respond by not enabling the spare processor to be initially allocated for OS and application processing.” *Id.*, [0034].

A POSITA would have understood that the disclosure in Fox relied upon by Petitioner is still performed during system initialization, but is not done “before the BIOS hands over control to the OS” which “may require some administrative input and support.” *Id.* In Fox, “[t]he spare processor remains hidden until one of the primary processors fails and the spare processor is needed to replace the failing processor.” *Id.*, [0037]. Fox teaches that its “invention provides a method for hot swap spare processors without the need of processor architecture changes, ...”. *Id.*,

[0047], and its claims recite that “the processing load of the failing processor is automatically sent to the spare processor for processing.” *Id.*, claim 1, claim 12.

A POSITA would not have disregarded the totality of Fox's teaches, but would instead understand that Fox only teaches a hot-spare processor. A POSITA would also understand that Fox does not teach idling an active spare processor. Fox only teaches idling a failing processor. In Fox, “[i]mplementation of the invention involves the utilization of Systems Management Interrupt (SMI) functionality provided within basic input output system (BIOS) of the IA-32 systems” and “[w]hen a processor failure is detected, SMI code running on a good processor instructs the OS to hold off allocating processes to the failing processor.” *Id.*, [0040]. “The SMI is generated by hardware during various processor failure conditions.” *Id.*, [0041]. Fox does not disclose, teach or suggest using the SMI idle a processor that is not failing, such as an active spare processor.

Petitioner relies on the same disclosure in Fox for claims 3, 4, 15 and 16, but, as explained above, a POSITA would not understand that Fox discloses, teaches, or suggests using an active spare instead of a hot-spare. Accordingly, Petitioner has not met its burden to establish that claims 3, 4, 15 and 16 are unpatentable.

G. Grounds 1 and 2 do not establish that challenged dependent claims 6, 8, and 25 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.

As noted above, Petitioner asserts that dependent claims 6, 8, and 25 would have been obvious to a POSITA over a combination of three or four references – (1) Fox, Safford, and Arai or (2) Fox, Safford, Arai and Landry. Paper 1, Pet., 59-62. Petitioner does not make separate arguments for Grounds 1 and 2. *Id.*

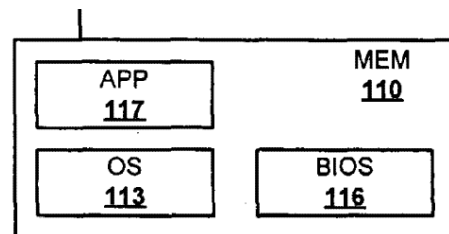
Petitioner's arguments fail because claims 6, 8 and 25 recite limitations requiring non-volatile data storage. Claim 6 recites "accessing information stored to non-volatile data storage that identifies one of the processors in said multi-processor system that is assigned the role of boot processor." Ex. 1001, '302 Patent, 19:37-38 (emphasis added). Claim 8 similarly recites "accessing information stored to non-volatile data storage that identifies one of the processors in said multi-processor system that is assigned the role of spare processor." *Id.*, 19:46-48 (emphasis added). Claim 25 similarly recites "non-volatile data storage storing information that assigns the role of system boot processor to one of said plurality of processor modules." *Id.*, 21:2-4 (emphasis added).

As Petitioner concedes, Fox does not disclose storing the information Petitioner alleges corresponds to the claimed information identifying "one of the processors" (*e.g.*, processor register or table) in system memory. Paper 1, Pet., 60.

As Petitioner further concedes, Arai discloses storing similar information (*e.g.*, ACPI or MP table) in main memory. *Id.* Petitioner then takes a logical leap, asserting that Fox's register or table would have been stored in Fox's system memory and that the system memory would be non-volatile. *Id.*

Petitioner's argument fails because it is based on a false premise – that Fox's system memory is “non-volatile” because a portion of what Fox identifies as system memory, BIOS, is commonly embedded in a non-volatile memory. *Id.* Petitioner even cites ROM (Read Only Memory) as the non-volatile memory in which BIOS would be embedded. *Id.* However, the system memory disclosed in Fox is not just BIOS. Fox discloses “a system memory 110, within which is stored software for controlling the MP and executing application processes. These software include: the OS 113, BIOS 116, which includes SMI, and program applications 117.” Ex. 1005, Fox, [0028]. *See also* Figure 1, partially reproduced below.

Figure 1



Id., 2, Figure 1. Fox teaches using “providing a software register of active processors in the OS and setting the register to reflect the spare processor as active and the failing processor as inactive.” *Id.*, [0041]. Fox does not teach providing a software

register in the BIOS. Instead, Fox claims “a processor register linked to said BIOS.” *Id.*, claims 2, 13. A POSITA would not make the mistaken assumption that all of the system memory in Fox is non-volatile memory just because some of the system memory, the BIOS, is non-volatile.

A POSITA would also understand that information identifying a boot processor or a spare processor would not be stored in memory that could not be written to, such as ROM, in the inventive methods or systems claimed in the '302 Patent. The '302 Patent discloses and claims transferring or switching the role of boot processor, something which could not be done with memory that cannot be written to.

Petitioner's citation to Ex. 1028, an unauthenticated apparent draft of MultiProcessor Specification, Version 1.4, (“MP Specification”) is unavailing even if the Board does not reject Petitioner's improper reliance on an unauthenticated draft publication that is not a reference identified in any of the four Grounds of the Petition. *See* Paper 1, Pet., 1, 61. The MP Specification explicitly teaches that “[t]hese two MP configuration data structures can be located in the ROM space only if the system is not dynamically reconfigurable.” Ex. 1028, MP Specification, 38. All of the claims of the '302 Patent are directed to a dynamically reconfigurable system in which the role of boot processor is switched or transferred to a spare

processor without shutting down the operating system (*i.e.*, dynamically reconfiguring the system).

Petitioner's arguments with respect to claim 25 are identical to those made for claim 6, (Paper 1, Pet., 62), and should be rejected for the same reasons set forth above. Petitioner's arguments with respect to claim 8 are also identical to those made for claim 6, (Paper 1, Pet., 65), and should be rejected for the same reasons set forth above.

Accordingly, Petitioner has not met its burden to establish claims 6, 8, or 25 are unpatentable.

H. Grounds 1 and 2 do not establish that challenged dependent claims 11, 12, 17 and 24 are rendered obvious by Fox in view of Safford and Arai or Fox in view of Safford, Landry and Arai.

As noted above, Petitioner did not present any arguments that claims 11, 12, 17, or 24 are unpatentable, even though they are included in the range of claims allegedly encompassed under Grounds 1 and 2. Paper 1, Pet., 1. The Federal Circuit has repeatedly reprimanded the Board for finding a claim unpatentable based on theories not presented in the petition. *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023) (citing *In re Magnum Oil Tools Int'l, Ltd.*, 829 F.3d 1364, 1380-81 (Fed. Cir. 2016); *In re NuVasive Inc.*, 841 F.3d 966, 968 (Fed. Cir. 2016); *Oren*

Techs., LLC v. Proppant Express Invs. LLC, 2021 U.S. App. LEXIS 21859, 2021 WL 3120819, at *4-*5 (Fed. Cir. July 23, 2021)).

As the Federal Circuit has held, “the Board should ... not have to decode a petition to locate additional arguments beyond the ones clearly made.” *Id.* “It is of the utmost importance that petitioners in the IPR proceedings adhere to the requirement that the initial petition identify ‘with particularity’ the ‘evidence that supports the grounds for the challenge to each claim.’” *Intelligent Bio-Systems, Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016) (quoting 35 U.S.C. § 312(a)(3) (2012)). This is required by the applicable Regulations governing *inter partes* review proceedings, 37 C.F.R. § 42.104(b)(4)-(5), requiring a petitioner to identify “[h]ow the construed claim is unpatentable,” “specify where each element of the claim is found in the prior art patents or printed publications relied upon,” and identify “[t]he exhibit number of the supporting evidence relied upon to support the challenge and the relevance of the evidence to the challenge raised, including identifying specific portions of the evidence that support the challenge.”

Nor can Petitioner's failure to argue these claims be cured in subsequent submissions. The Federal Circuit has “rejected, many times, post-hoc attempts on appeal to include additional, new arguments not contained in the petition.” *Netflix, Inc. v. DivX, LLC*, 84 F.4th 1371, 1377 (Fed. Cir. 2023) (citing *Microsoft Corp. v.*

Biscotti, Inc., 878 F.3d 1052, 1074 (Fed. Cir. 2017); *Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.*, 853 F.3d 1272, 1286 (Fed. Cir. 2017) (“We also are unpersuaded by Continental's attempts to cure the petition's deficiencies in its subsequent briefing to the Board and to us.”)).

Accordingly, there is no basis for finding claims 11, 12, 17, or 24 are unpatentable and this Board cannot properly do so in this *inter partes* review proceeding.

VII. Conclusion

Petitioner has failed to meet its burden of establishing that the Challenged Claims are unpatentable and the Petition should be denied for all claims.

Respectfully submitted,

Dated: June 20, 2025

/s/ Brett Cooper
Brett Cooper, Reg. No. 55,085
Lead Counsel for Patent Owner
BC Law Group, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016
(212) 951-0100

CERTIFICATE OF WORD COUNT (37 C.F.R. § 42.24(d))

The undersigned hereby certifies that the word count in this Patent Owner's Response is 10,797, as counted by the word-processing program Microsoft Word that was used to generate this Patent Owner's Response, where such word count excludes the table of contents, table of authorities, mandatory notices, certificate of service, appendix of exhibits, claim listing, and this certificate of word count. This Patent Owner Response is in compliance with the 14,000-word limit set forth in 37 C.F.R. § 42.24(b)(1).

Date: June 20, 2025

/Brett Cooper/

BC Law Group, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016
(212) 951-0100

Brett Cooper
Reg. No. 55,085
Attorney for Patent Owner

CERTIFICATE OF SERVICE (37 C.F.R. § 42.6(e))

The undersigned hereby certifies that the above document was served on June 20, 2025, by filing this document through the Patent Trial and Appeal Case Tracking System as well as delivering a copy via electronic mail upon the following attorneys of record for the Petitioner:

Lionel M. Lavenue (Reg. No. 46,859)
lionel.lavenue@finnegan.com
Deanna Smiley (Reg. No. 79,284)
deanna.smiley@finnegan.com
Finnegan, Henderson, Farabow,
Garrett & Dunner, LLP
1875 Explorer Street, Suite 800
Reston, VA 20190-6023

Kara A. Specht (Reg. No. 69,560)
kara.specht@finnegan.com
Finnegan, Henderson, Farabow,
Garrett & Dunner, LLP
271 17th Street, NW, Suite 1400
Atlanta, GA 30363-6209

Cory C. Bell (Reg. No. 75,096)
cory.bell@finnegan.com
Xirui Zhang (Reg. No. 78,809)
xirui.zhang@finnegan.com
Finnegan, Henderson, Farabow,
Garrett & Dunner, LLP
2 Seaport Lane, 6th Floor
Boston, MA 02210-2001

Alissa E. Green (Reg. No. 78,501)
alissa.green@finnegan.com

Finnegan, Henderson, Farabow,
Garrett & Dunner, LLP
901 New York Avenue, NW
Washington, DC 20001-4413
Tel: 202-408-4000
Fax: 202-408-4400

BMW-Foras-IPR@finnegan.com

Date: June 20, 2025

/Brett Cooper/

BC Law Group, P.C.
200 Madison Avenue, 24th Floor
New York, NY 10016
(212) 951-0100

Brett Cooper
Reg. No. 55,089
Attorney for Patent Owner