

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 IPR2024-01346
U.S. Patent No. 7,627,781

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.	Challenged Claims.....	1
III.	Background.....	2
IV.	Person of Ordinary Skill in the Art.....	3
V.	Response to Statement of Facts.....	3
VI.	Ground 1	6
	A. Petitioner does not establish in Ground 1 that claim 1 is rendered obvious by Fox in view of Safford and Arai	6
	1. Limitation [1b] “data storage storing information that assigns a role of boot processor to one of said plurality of processors”	6
	2. Limitation [1c] “logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor”	9
	B. Petitioner does not establish in Ground 1 that claim 2 is rendered obvious by Fox in view of Safford and Arai	11
	C. Petitioner does not establish in Ground 1 that claim 14 is rendered obvious by Fox in view of Safford and Arai	14
VII.	Ground 2	16
	A. Limitation [1b]: “ <i>data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors</i> ”	17

B.	Limitation [1c]: “logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor”	18
C.	Claim 2.....	19
VIII.	Grounds 1 and 2 – The proposed combinations of Fox, Safford, and Arai (Ground 1) and Fox, Safford, Arai, and Landry (Ground 2) would not render independent claims 9, 16, or 25 obvious to a POSITA.....	19
IX.	Grounds 3 and 4 – The proposed combinations of Fox, Safford, Arai, and Jin (Ground 3) and Fox, Safford, Landry, Arai, and Jin (Ground 4) would not render obvious claims 3-4, 19-21, or 27-28.	21
X.	Grounds 5 and 6 – The proposed combinations of Fox, Safford, Arai, and Bigbee (Ground 5) and Fox, Safford, Landry, Arai, and Bigbee (Ground 6) would not render obvious claims 8 or 24.	21
XI.	Conclusion.....	21

TABLE OF AUTHORITIES

Cases

<i>Stryker Corp. v. Karl Storz Endoscopy America, Inc.</i> , IPR2015-00677, Paper 15 (PTAB Sept. 2, 2015).....	7
---	---

Rules

37 C.F.R. § 42.....	3
---------------------	---

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	Deposition Transcript of Dr. Michael Brogioli

I. Introduction

Patent Owner respectfully requests that the Board find all of Petitioner's challenges to the claims of U.S. Patent No. 7,627,781 ("the '781 patent") without merit. First, none of the prior art, separately or combined, includes or suggests transferring, during system runtime, the role of boot processor to a spare processor. Second, Petitioner repeatedly relies on two non-prior art references for claim limitations missing from the alleged prior art asserted as a basis for the grounds in the Petition. Third, Petitioner's obviousness challenge to claim 2 fail because the prior art cited by Petitioner does not disclose, teach or suggest a "*non-volatile data storage*" which is a limitation of this claim.

II. Challenged Claims

In Ground 1, Petitioner challenges patentability of claims 1-2, 5-7, 9-18, 22-23, 25-26, and 29-30 of the '781 patent based on Fox in view of Safford and Arai. Petitioner challenges the same set of claims under Ground 2 based on Fox in view of Safford, Landry, and Arai. In Ground 3, Petitioner challenges claims 3-4, 19-21, and 27-28 in view of Safford, Arai, and Jin. Petitioner challenges the same set of claims under Ground 4 based on Fox in view of Safford, Landry, Arai, and Jin. In Ground 5, Petitioner challenges claims 8 and 24 based on Fox in view of Safford, Arai, and Bigbee. And under Ground 6, Petitioner challenges claims 8 and 24 based on Fox in view of Safford, Landry, Arai, and Bigbee.

III. Background

The '781 patent is titled “System and Method for Establishing a Spare Processor for Recovering from Loss of Lockstep in a Boot Processor” and was filed on October 25, 2004. *See* Ex. 1001.

The '781 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.*, 3:10-13. As noted in the '781 patent, “crashing the system each time LOL is detected is not an attractive proposition,” especially for large systems having many processors. *Id.*, 3:17-24. The '781 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 '781 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.*, 5:7-10. The '781 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.*, 5:10-16.

IV. Person of Ordinary Skill in the Art

A person of ordinary skill in the art in the field of the '781 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 '781 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* EX2004, 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 ("Pet.") at 4. None of the prior art relied upon by Petitioner would have rendered any of the claims of the '781 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. Response to 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. Ground 1

A. Petitioner does not establish in Ground 1 that claim 1 is rendered obvious by Fox in view of Safford and Arai

Petitioner has not shown that the Fox-Safford-Arai combination renders obvious limitation [1b] “*data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors*” or [1c] “*transferring, during system runtime, the role of boot processor to said spare processor.*”

1. Limitation [1b] “*data storage storing information that assigns a role of boot processor to one of said plurality of processors*”

Petitioner contends that “Fox discloses a multiprocessor system configured with IA-32 architecture” and that “[t]herefore 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. Petitioner further contends that “[a] POSITA would have understood that Fox’s IA-32 system would have set the BSP flag of the designated boot processor to 1 (‘assigns a role of boot processor’)” and that “in an IA-32 multiprocessor system, like that in Fox, the identification and software flags of processors, including those of the boot processor and the spare processor (‘information’), would be stored in Fox’s register or table created by the boot processor and accessible to the BIOS and operating system.” Pet. 22-23. Thus, Petitioner identifies a “BSP flag” as the alleged “*information that assigns a role of*

boot processor” in claim 1 and identifies “Fox’s register or table created by the boot processor” as the alleged “*data storage*.” There are several deficiencies in Petitioner’s arguments.

First, as Petitioner’s expert acknowledged, Fox does not disclose a boot processor or the BSP flag Petitioner relies upon for the “*assigns a role of boot processor*” limitation. Indeed, 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. EX2006 at 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. EX2006 at 30:5-31:24. A copyright date, by itself, is insufficient evidence of a document’s date of publication, as it provides no indication that the document was publicly accessible as of that date. *Stryker Corp. v. Karl Storz Endoscopy America, Inc.*, IPR2015-00677, Paper 15, 18-19 (PTAB Sept. 2, 2015).

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 ’781 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. For these reasons, the disclosures in Exhibit 1008 relating to a BSP flag are insufficient to demonstrate obviousness of the “*assigns a role of boot processor*” limitation at the time of the invention of the ’781 patent.

The Petition is similarly deficient with respect to EX1028, referred to by Petitioner as “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.” EX2006 at 34:10-35:9. Dr. Brogioli’s speculation that the admitted draft version of EX1028 was “made available” is uncorroborated and based on unsupported speculation regarding “various versions of these types of documents.” EX2006 at 36:1-20. Petitioner presents no evidence that EX1028 was ever publicly accessible. For these reasons the Board should give Petitioner’s reliance on Exhibits 1008 and 10028 no weight.

Second, Petitioner contends that the registers or tables in Fox that purportedly store flags such as a boot processor flag, are “created by the boot processor.” Pet.

22-23. Yet claim 1 recites “*data storage storing information that assigns a role of boot processor,*” and Petitioner has provided no explanation as to how a role of boot processor can be assigned by a boot processor itself. Such circular reasoning is flawed and does not demonstrate that the limitations of claim 1 are obvious.

2. Limitation [1c] “logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor”

Petitioner’s arguments in Ground 1 as to this limitation are flawed. Petitioner contends that “*transferring, during system runtime, the role of boot processor to said spare processor*” is obvious based on Fox’s disclosure that “during the processor switching, the executing threads/processes are transferred to the replacement processor, and the memory save state is switched from the failing processor to the spare processor” and Arai’s disclosure of “switching the logical APIC IDs of the active processor in error with the reserved processor.” Pet. 28. Petitioner contends that a POSITA would have understood that switching logical APIC IDs would have “changed the logical APIC ID of the processor in error to the logical APIC ID of the reserved processor.” Pet. 29. But Petitioner has not shown that switching the logical APIC IDs between processors has any impact on which processor performs any boot processor role. Jones ¶ 34. Therefore, Petitioner has not shown that the “role of boot processor” is transferred or that it would be obvious to do so.

Furthermore, as Mr. Jones confirms, the APIC IDs in Arai are stored in RAM. Jones ¶ 35; Ex. 1007 at ¶ 20 (“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”), ¶ 21 (“The BIOS creates a table in the main memory (RAM) during start-up”). Dr. Brogioli also acknowledged during deposition that Arai’s table designating active/idle processors is stored in RAM. EX2006 at 103:14-21. And, as to the Fox-Arai combination, Dr. Brogioli stated 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. EX2006 at 108:10:10-109:4; see *also id.* at 110:4-11, 111:12-112:12. As Mr. Jones states, “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.’” Jones ¶ 36.

Furthermore, although claim 1 recites a transfer that occurs “*during runtime*,” the specification and plain language of the claim show that the “*role of boot*

processor” relates to what occurs in the system during the boot process. The ’781 specification discloses, for example, that changes to processor roles can change settings in NVRAM that are “honored on the next reset.” EX1001 14:4-31. Also, in the description of the “device tree,” the ’781 specification describes that a processor identified as a “hot spare” in the device tree does not show up in the static ACPI tables that are built when the system boots. It then says that “upon such spare assuming the role of boot processor, its status is updated to ‘Present, Enabled, and Functioning’.” EX1001, 13:36-55. This suggests that updating the status is a separate step from transferring the role. Column 17:45-55 says that “the firmware assigns active spare processor the role of boot processor” and “the firmware updates the device tree to identify the active spare as the boot processor, and performs any other configuration changes (e.g., changing the LID of the active spare, etc.) to cause the OS to recognize the active spare as the boot processor,” which demonstrates that assigning the **role of boot processor** is more than merely: (1) copying the **state of the boot processor** into the spare, (2) updating the **device tree** to identify the spare as the boot processor, or (3) changing the **LID** of the spare processor.

B. Petitioner does not establish in Ground 1 that claim 2 is rendered obvious by Fox in view of Safford and Arai

Petitioner has not shown that the Fox-Safford-Arai combination renders obvious claim 2. Petitioner contends that Fox-Arai renders obvious storing Fox’s

register or table in its system memory and that “Fox’s system memory would be ‘non-volatile’ because it was common and well known that the BIOS would be embedded in a non-volatile memory (e.g., ROM).” Pet. 50. This is insufficient because the alleged “data storage” in the Fox-Arai combination is RAM, which is volatile memory.

The storage location of the BIOS is irrelevant to the “*data storage storing information*” limitation because the BIOS is not the alleged “*information*” in claim 1. Rather, Petitioner identifies Fox’s “identifications and software tags of the processors” as “*information*” that is “stored in Fox’s register or table.” Pet. 36. And, in the alleged Fox-Arai combination, the register or table are **created by** the BIOS and stored in volatile RAM. Pet. 23 (“Arai discloses that during system initialization, the BIOS creates a table (e.g., ACPI table or MP table) that identifies each processor in the system and its designated operational state, i.e., active or disabled. Arai discloses this the table is stored in main memory (‘data storage’).”). As Dr. Brogioli confirmed, in both Arai and the Fox-Arai combination, the identified table or register is stored in RAM. EX2006 at 111:12-112:12 (“That’s what’s happening in the combination is, just as Arai says in paragraph 21 . . . that the BIOS goes and creates a table in that main memory during the start-up in the RAM. So, the BIOS is coming out of this read only memory, running and creating the table in the RAM here.”);

EX1007 ¶ 20-21; Jones ¶ 35; *see also* EX2006 at 103:14-21, 108:10:10-109:4, 110:4-11.

Thus, even if the BIOS in the Fox-Arai combination is itself stored in non-volatile memory, as Petitioner contends, the table or register allegedly comprising “*data storage storing information that assigns a role of boot processor*” is expressly stored in volatile memory, and there is no disclosure that such information is stored elsewhere. Accordingly, Petitioner has not shown that the Fox-Arai combination discloses or renders obvious “*data storage storing information that assigns a role of boot processor*” (claim 1) “*wherein said data storage is non-volatile data storage*” (claim 2).

Petitioner’s contention that “[t]he ’781 Patent similarly discloses a device tree or ACPI table that is stored in a RAM, accessible to the firmware, and designates which processor is the boot processors” is unpersuasive. Pet. 24 (citing EX1001, 8:43-54, 12:59-65, 13:20-44). It is unclear how this disclosure relates to the “*non-volatile data storage*” limitation in claim 2, as there is no suggestion that RAM is non-volatile data storage, and claim 2 does not prohibit storage of information elsewhere. Indeed, although the ’781 specification describes embodiments where an ACPI table is stored in SRAM, it also includes embodiments where, for example, information regarding the boot device path can be changed during runtime and persistently stored “for the next time the system reboots.” Ex. 1001 at 12:54:56; *see*

also id. at 12:35-58, 8:4-31; Jones ¶ 37. The '781 specification further discloses, for example, an embodiment where “commands for assigning the boot processor and/or spare processor” are pushed “into NVRAM so they will be honored on the next reset.” Ex. 1001 at 14:4-32.

C. Petitioner does not establish in Ground 1 that claim 14 is rendered obvious by Fox in view of Safford and Arai

Claim 14 recites “*during system boot-up, system firmware identifying the spare processor as an active spare.*” In the Petition, Petitioner identifies Fox’s disclosure in paragraphs [0015] and [0034]. Pet. 56. The Petition says “Fox discloses during system BIOS boot (‘during system boot-up’), the BIOS (‘system firmware’) reserving the spare processor in an idle state and identifies the spare processor by setting a software tag.” Pet. 56 (emphasis added). However, the description of an “active spare” processor in the '781 patent does not include a processor in an idle state. The '781 patent describes that “a processor is designated as a spare processor for the boot processor, but such designated spare processor is not held in reserve (idled) during normal system operation. Rather, the designated spare is utilized as an **active spare** processing resource made available to OS 11 for processing instructions.” EX1001, 7:35-40 (emphasis added). The specification also distinguishes a “hot spare” from an “active spare” when it states “the firmware determines . . . the processor designated as the active spare, and the firmware

instructs the OS to ‘eject’ the active spare processor. Thus, the active spare is idled and effectively transformed into a hot spare.” EX1001, 17:41-46.

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.” EX1005, [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.

Accordingly, Petitioner's contention that Fox's disclosure of a processor in an idle state is insufficient to render obvious an "active spare." In addition, when asked during his deposition about the difference between a "hot spare" and an "active spare," Petitioner's expert admitted that his interpretation of Fox is that an "active spare" is a "hot-spare" that has been added to the active list. EX2006 at 78:4-16; *see also id.* at 62:7-78:3. This is also inconsistent with the claim language, which recites that the identification of the "*active spare*" is done "*during system boot-up*." Petitioner has failed to show that Fox's "hot spare" is an "*active spare*" as recited in claim 14, because Petitioner points to no disclosure in Fox where, during system boot-up, the system firmware identifies a processor as a spare processor and, during system boot-up, places that processor on the active list.

For these reasons, Petitioner has not met its burden as to claim 14.

VII. Ground 2

In Ground 2, Petitioner alleges that the Fox-Safford-Arai combination would have further rendered claim 1 obvious in view of Landry. In particular, Petitioner contends that "a POSITA would have understood Landry's process as a suitable option for selecting a boot processor in Fox's system consistent with the initialization protocol of an IA-32 multiprocessor system." Pet. 34.

A. Limitation [1b]: “*data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors*”

As to limitation [1b], which recites “*data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors,*” Petitioner alleges that “Fox in view of Landry further renders obvious assigning a role of boot processor—logical CPU0—to one of said plurality of processors.” Pet. 36. As to the “data storage” element, Petitioner relies, as it did under Ground 1, on Fox’s register or table, which would be stored in system memory “like in Arai.” Regardless of the logical identification given to each processor—e.g., “logical CPU0”—Petitioner’s argument as to this limitation in Ground 2 fails for the same reasons it failed in Ground 1. Petitioner has not shown that the “identifications and software tags” that are stored in Fox’s register or table (in RAM) have any impact on which processor performs the role of boot processor.

In addition, to the extent that Petitioner contends that Fox would be modified to implement Landry’s boot processor selection process, Petitioner has not provided any evidence or argument that such a process includes “*data storage for storing information that . . . assigns a role of spare processor to another of said plurality of processors,*” as recited in claim 1.

B. Limitation [1c]: “logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor”

Petitioner contends that the Fox-Landry-Safford-Arai combination renders obvious “*logic operable, responsive to detecting loss of lockstep for said boot processor, for transferring, during system runtime, the role of boot processor to said spare processor.*” Pet. 37. Petitioner argues that “transferring the role of the boot processor to the spare processor would include switching the logical CPU0 of the boot processor to the spare processor, as taught in Safford and Arai.” Pet. 37. This contention is unsupported because, as discussed in Ground 1, there is no evidence that switching the logical identifiers—e.g., Arai’s APIC ID, or Landry’s logical CPU identifier—between processors “*transfer[s], during system runtime, the role of boot processor to said spare processor.*”

In Safford, there is no mention of any boot processor. And in Arai, “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 transferred.” Jones ¶ 43. In addition, as explained in reference to Ground 1, the designations in Arai are written to volatile RAM and do not persist on reboot.

In addition, Petitioner’s reliance on Landry is insufficient. Petitioner contends that “Fox in view of Landry further renders obvious assigning a role of boot

processor—logical CPU0—to one of said plurality of processors.” Pet. 36. But Landry only discloses assigning processor roles during system initialization, which occurs before system runtime. EX1002 ¶ 147; *see* EX2006 at 87:21-88:15. In addition, as Dr. Brogioli acknowledged during his deposition, the designation of a boot processor in Landry is only retained until the “system is power cycled” or rebooted, at which point Landry’s self-testing process is repeated. EX1011 at 6 (col 2:20-23); EX2006 at 88:16-90:16. Accordingly, Landry does not support Petitioner’s argument regarding transferring “*during system runtime*” the role of boot processor to a spare processor.

C. Claim 2

Petitioner presents no independent argument for claim 2 based on the Fox-Landry-Safford-Arai combination. Thus, it has not shown claim 2 is obvious under Ground 2 for the same reasons it failed under Ground 1. Pet. 50.

VIII. Grounds 1 and 2 – The proposed combinations of Fox, Safford, and Arai (Ground 1) and Fox, Safford, Arai, and Landry (Ground 2) would not render independent claims 9, 16, or 25 obvious to a POSITA.

Grounds 1 and 2 also challenge independent claims 9, 16, and 25, largely repeating the arguments presented as to claim 1.

As to claim limitation [9c], “*responsive to detecting loss of lockstep for said boot processor, changing, during system runtime, the assignment of the role of boot processor to said spare processor;*” Petitioner alleges obviousness for the same

reasons explained in Grounds 1 and 2 for limitation [1c]. Accordingly, this argument fails for the same reasons previously provided for Grounds 1 and 2 as to limitation [1c].

As to limitation [16d], “*data storage storing information that assigns a role of boot processor to one of said plurality of processors and assigns a role of spare processor to another of said plurality of processors,*” Petitioner relies entirely on its prior arguments in Grounds 1 and 2 for limitation [1b]. These arguments fail for the same reasons previously provided for Grounds 1 and 2 as to limitation [1b].

As to limitation [25f], “responsive to detecting loss of lockstep for said boot processor, said system causing said spare processor to dynamically assume the role of boot processor,” Petitioner alleges obviousness for the same reasons set forth with respect to limitation [1c] in Grounds 1 and 2. Accordingly, this argument fails for the same reasons previously provided for Grounds 1 and 2 as to limitation [1c].

Because Petitioner has not shown any of the independent claims of the ’781 patent is obvious under Grounds 1 or 2, its challenge to dependent claims 2, 5-7, 10-18, 22-23, 26, and 29-30 must also fail. In addition, Petitioner makes no independent argument for claim 14 under Ground 2, so that claim is non-obvious for the same reasons addressed above as to claim 14 under Ground 1.

IX. Grounds 3 and 4 – The proposed combinations of Fox, Safford, Arai, and Jin (Ground 3) and Fox, Safford, Landry, Arai, and Jin (Ground 4) would not render obvious claims 3-4, 19-21, or 27-28.

Petitioner's challenges under Grounds 3 and 4 to dependent claims 3-4, 19-21, and 27-28 fail for the same reasons provided above for independent claims 1, 16, and 25, on which they depend. Petitioner presents no independent analysis of the limitations of independent claims 1, 16, or 25 for either Ground 3 or 4. Pet. 63-64.

X. Grounds 5 and 6 – The proposed combinations of Fox, Safford, Arai, and Bigbee (Ground 5) and Fox, Safford, Landry, Arai, and Bigbee (Ground 6) would not render obvious claims 8 or 24.

Petitioner's challenges under Grounds 5 and 6 to dependent claims 8 and 24 fail for the same reasons provided above for independent claims 1 and 16, on which they, respectively, depend. Petitioner presents no independent analysis of the limitations of independent claims 1 and 16 for either Ground 5 or 6. Pet. 66.

XI. 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.

Date: June 20, 2025

Respectfully submitted,

/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 5,013, 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)(1))

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. 55,085
Attorney for Patent Owner